

Computer Networks

Kamil NOWAK

Department of Computer Science

Faculty of Computer Science and Management at Wrocław University of Technology

kamil.nowak@pwr.edu.pl

Transport Layer of the ISO-OSI model

- Transmission Control Protocol (TCP)

- User Datagram Protocol (UDP)

Version 0.0-190423 transl. 191105

Keywords

Transport protocol, TCP, Transmission Control Protocol, User Datagram Protocol, UDP, segment, datagram, TCP header, UDP header, Transport protocol ports

Introduction

Protokoły działające w warstwach modelu ISO-OSI w celu umożliwienia działania sieci komputerowej muszą współpracować między sobą. Warstwy sąsiednie świadczą dla siebie nawzajem usługi. Warstwa transportowa musi współpracować z warstwą sieciową i warstwą sesji (w modelu TCP/IP warstwą aplikacji).

Najpopularniejszy na świecie protokół sieciowy IP do poprawnej pracy potrzebuje współdziałania z innymi protokołami warstwy sieciowej i sąsiednich. Te współpracujące protokoły nazwane są stosem protokołów TCP/IP. Większość protokołów z tej grupy jest ściśle zdefiniowana i nie może być zastąpiona innymi, jakby to wynikało z otwartego modelu warstwowego sieci komputerowych. Do poprawnego działania warstwy sieciowej potrzebna jest np. współpraca z protokołem ICMP. Do poprawnej komunikacji z warstwą aplikacja protokół IP korzysta z dwóch predefiniowanych protokołów transportowych:

- TCP – Transport Control Protocol;
- UDP – User Datagram Protocol.

Protokół TCP jest protokołem połączeniowym, który zapewnia dostarczenie danych od odbiorcy do nadawcy w niezmienionym stanie. Protokół UDP jest protokołem bezpołączeniowym, jest szybszy, o mniejszym narzucie danych sterujących, ale nie daje gwarancji poprawności dostarczenia wysyłanych przez nadawcę danych. Protokół UDP nie gwarantuje również, że wysłane dane dotrą do odbiorcy w odpowiedniej kolejności.

Spis treści

Protokoły transportowe stosu TCP/IP

Opis protokołu TCP

Nagłówek protokołu TCP

Opis protokołu UDP

Nagłówek protokołu UDP

Analiza protokołów TCP i UDP w programie Wireshark

Analiza protokołu TCP

Uruchamiając program Wireshark bardzo łatwo jest w nim znaleźć ramki z protokołem TCP. Większość aplikacji komunikuje się między sobą używając właśnie tego protokołu transportowego. Otwierając strony serwisu WWW, używając programu telnet, czy SSH, kopiując pliki z serwera FTP generujemy dane dla protokołu TCP. Aby przeanalizować całą sesję i działanie protokołu TCP wygodnie jest posłużyć się aplikacją FTP (File Transfer Protocol). Aplikacja ta służy do kopiowania plików przez sieć. Mamy więc możliwość przesyłania dowolnej ilości danych w postaci pliku tekstowego lub binarnego i zaobserwowania procesu wymiany danych pomiędzy nadawcą i odbiorcą. Komputery używające protokołu FTP muszą mieć ze sobą łączność w sieci IP. Do celów ćwiczenia można użyć topologii bazowej sieci laboratoryjnej: dwóch komputerów połączonych poprzez przełącznik.

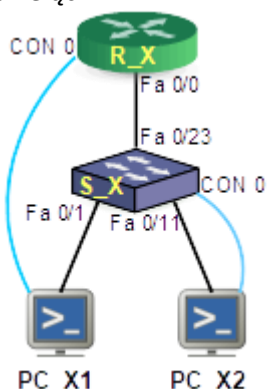
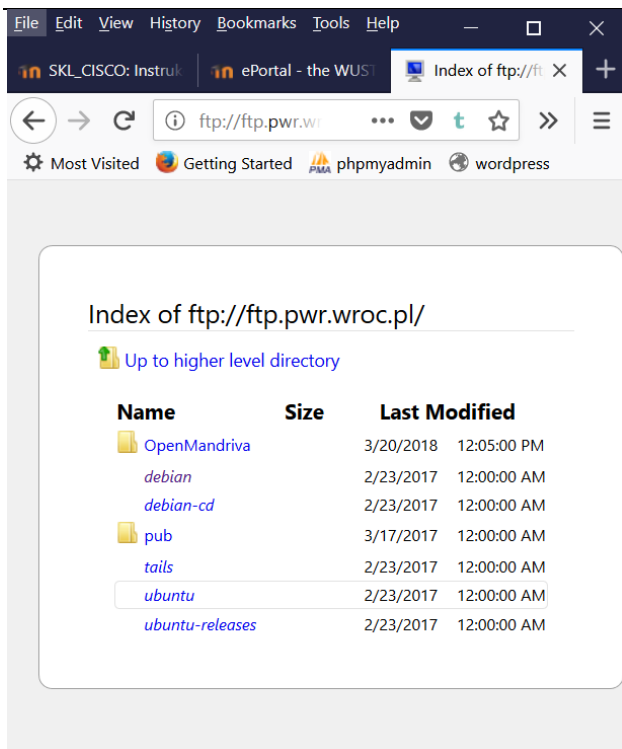


Figure 1 - Base topology of a single laboratory group.

Jeden z komputerów powinien pełnić rolę serwera plików FTP, a drugi klienta FTP. Aplikacja klienta i serwera FTP ma wiele implementacji. Możemy użyć aplikacji wbudowanych w system Windows, Linux lub inny. Możemy też zainstalować zewnętrzną aplikację o odpowiadającym nam interfejsie i funkcjach. Nie wszyscy zdają sobie sprawę, że przeglądarka internetowa również potrafi komunikować się w tym protokole. W oknie adresu URL (Uniform Resource Locator) adres serwera FTP należy poprzedzić nazwą protokołu: np. `ftp://ftp.pwr.wroc.pl`



Rysunek 2 - Przeglądarka internetowa pełniąca rolę klienta FTP.

Przed przesłaniem pliku należy skonfigurować usługę. Przykładowo skonfigurujemy komputer PC_X1 jako klienta FTP, a komputer PC_X2 jako serwer FTP. Na komputerze klienta przygotowujemy plik do przesłania na serwer. Można np. użyć katalogu c:\ftp i stworzyć w nim plik klient_ftp.txt. Wewnątrz pliku wpisujemy rozpoznawalną nazwę (np. imię i nazwisko twórcy pliku). Podobnie postępujemy na serwerze. W wypadku serwera katalog c:\ftp musi być skonfigurowany jako katalog główny serwera, w przeciwnym wypadku dodane do niego pliki nie będą widoczne przez klienta. Skonfigurowanie serwera FTP należy wykonać zgodnie z dokumentacją. W przypadku serwera FTP wbudowanego w MS Windows Server, należy dodać go jako usługę i skonfigurować.

```

Command Prompt - ftp ftp.pwr.edu.pl
c:\Users\tmp>
c:\Users\tmp>cd c:\ftp

c:\ftp>dir
Volume in drive C is WIN10
Volume Serial Number is 0BE9-0DB3

Directory of c:\ftp

22.03.2018  14:44    <DIR>        .
22.03.2018  14:44    <DIR>        ..
22.03.2018  14:44                0 klient_ftp.txt.txt
               1 File(s)                0 bytes
               2 Dir(s)  27 361 509 376 bytes free

c:\ftp>ftp ftp.pwr.edu.pl
Unknown host ftp.pwr.edu.pl.
ftp> open ftp.pwr.wroc.pl
Connected to ftp.pwr.wroc.pl.
220  .. :: Welcome on ftp.pwr.wroc.pl mirror server, provided by Wrocław
Centre of Networking and Supercomputing :: ..
200 Always in UTF8 mode.
User (ftp.pwr.wroc.pl:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp>

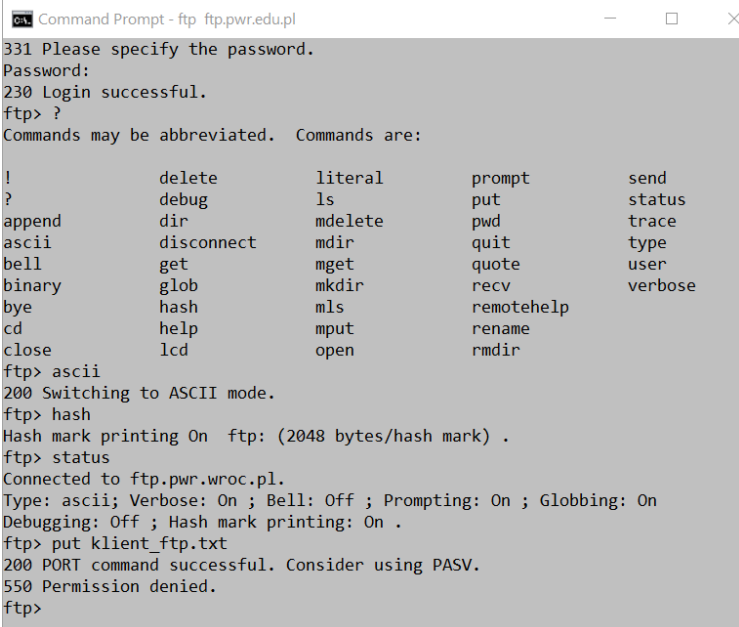
```

Figure 3- Establishing an ftp connection via the built in ftp client from the Windows cmd console.

Na komputerze klienta uruchamiamy okno cmd, zmieniamy katalog na zawierający nasz plik, a następnie wywołujemy polecenie ftp wraz z adresem serwera.

Serwer będzie wymagał od nas autoryzacji. Możemy zalogować się na znane, istniejące konto (np. administratora) lub użyć konta anonimowego (anonymous), jeżeli jest skonfigurowane na serwerze. Jako hasło podajemy wówczas swój e-mail.

Protokół FTP posiada dwa tryby transmisji: tekstowy i binarny. Tryb tekstowy wykorzystuje się przy kopiowaniu plików tekstowych pomiędzy różnymi systemami operacyjnymi. W różnych systemach operacyjnych koniec linii może być oznaczany w różny sposób. Np. w systemie Linux koniec linii oznaczany jest przez LF (Line Feed), a w systemie Windows przez CR+LF (Carriage Return + Line Feed). Bez konwersji wygląd plików w różnych systemach będzie się różnić.



```

C:\> Command Prompt - ftp ftp.pwr.edu.pl
331 Please specify the password.
Password:
230 Login successful.
ftp> ?
Commands may be abbreviated.  Commands are:

!            delete        literal        prompt        send
?            debug         ls            put           status
append       dir            mdelete       pwd           trace
ascii        disconnect    mdir          quit          type
bell         get           mget          quote         user
binary       glob          mkdir         recv          verbose
bye          hash          mls           remotehelp
cd           help          mput          rename
close       lcd           open          rmdir

ftp> ascii
200 Switching to ASCII mode.
ftp> hash
Hash mark printing On  ftp: (2048 bytes/hash mark) .
ftp> status
Connected to ftp.pwr.wroc.pl.
Type: ascii; Verbose: On ; Bell: Off ; Prompting: On ; Globbing: On
Debugging: Off ; Hash mark printing: On .
ftp> put klient_ftp.txt
200 PORT command successful. Consider using PASV.
550 Permission denied.
ftp>

```

Rysunek 4 - Parametryzacja aplikacji Windows klient ftp. Wysłanie pliku na serwer.

W celu pobrania plików użyjemy komendy get lub mget (pobieranie wielu plików). W celu wysłania plików na serwer użyjemy komendy put lub mput.

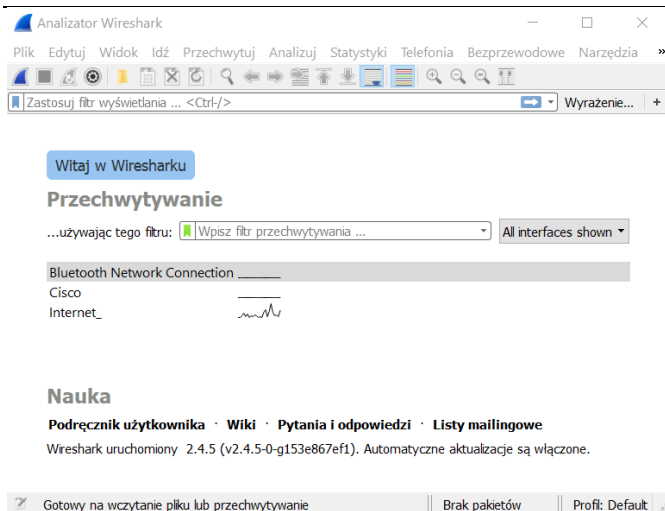
Do wyświetlenia zasobów serwera użyjemy komendy dir lub ls.

Do wykonania komendy lokalnej użyjemy znaku wykrzyknika np. !dir listuje zawartość lokalnego katalogu.

Aby zakończyć działanie usługi używamy komendy bye lub quit.

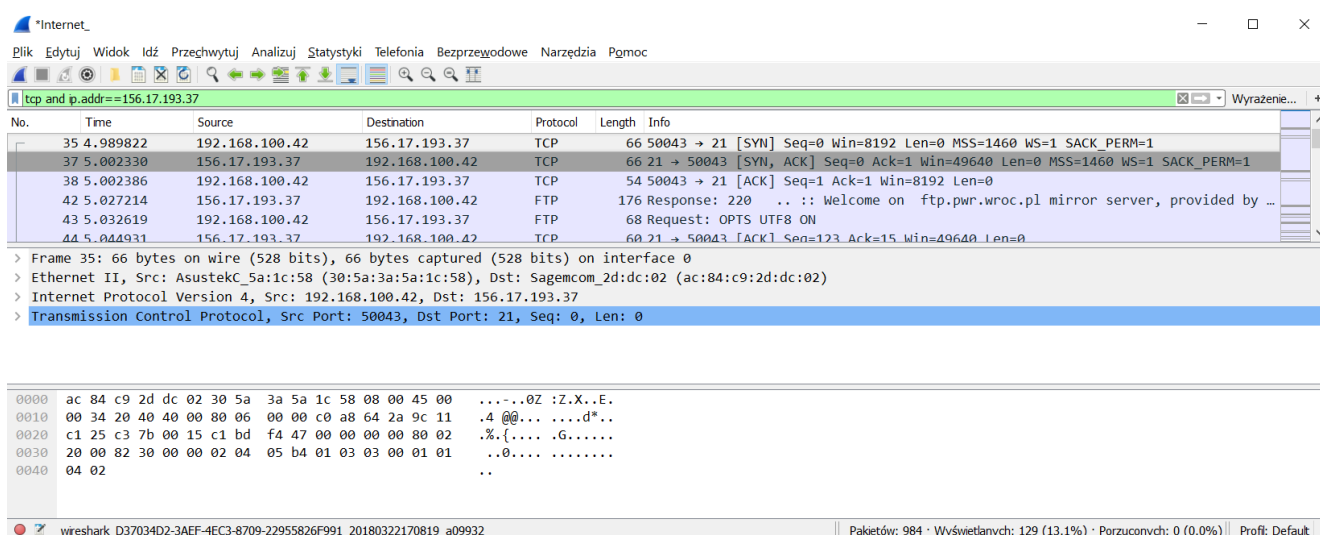
W celu przechwycenia sesji TCP wywołanej na potrzeby aplikacji FTP należy wykonać następujące czynności:

1. Przygotować pliki do przesłania.
2. Otworzyć okno Windows cmd i przygotować komendę ftp.
3. Uruchomić program Wireshark. Wybrać nasłuchiwanie na interfejsie, poprzez który będzie przesyłany plik.
4. Uruchomić klienta ftp. Zalogować się na serwer. Wylistować zdalny katalog.
5. Pobrać plik z serwera.
6. Wysłać plik na serwer.
7. Zakończyć działanie klienta FTP.
8. Zatrzymać nasłuchiwanie sieci przez program Wireshark.
9. Przeanalizować zeskanowane ramki.



Rysunek 5 - Okno programu Wireshark. Przy starcie użytkownik wybiera interfejs nasłuchujący.

Przy starcie aplikacji Wireshark program prosi o wybór interfejsu, na którym będzie nasłuchiwał. Parametr ten można następnie zmienić w trakcie pracy programu. Jest to dosyć ważny moment, gdyż zdarza się, że nasłuchujemy na interfejsie przez który nie przechodzi nasz ruch.



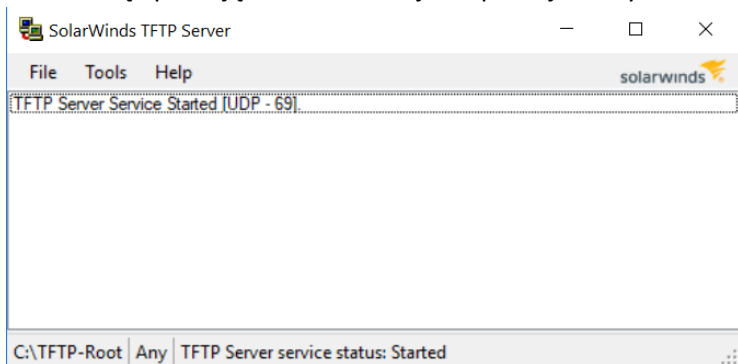
Rysunek 6 - Okno programu Wireshark. Przefiltrowany fragment danych połączenia ftp.

Powyższy rysunek pokazuje przykładowy listing ramek wygenerowany w czasie połączenia FTP. Pierwsze trzy ramki odpowiadają nawiązaniu sesji TCP – Three Way Handshaking. Transfer małego pliku tekstowego nie będzie wystarczający do pełnego przeanalizowania działania protokołu TCP. Plik może zmieścić się w jednej ramce. W celu zaobserwowania wielokrotnej transmisji danych bez potwierdzenia, potwierdzeń, działania okna TCP, potrzebne będzie przekopiowanie większego pliku o wielkości kilkudziesięciu megabajtów.

Analiza protokołu UDP

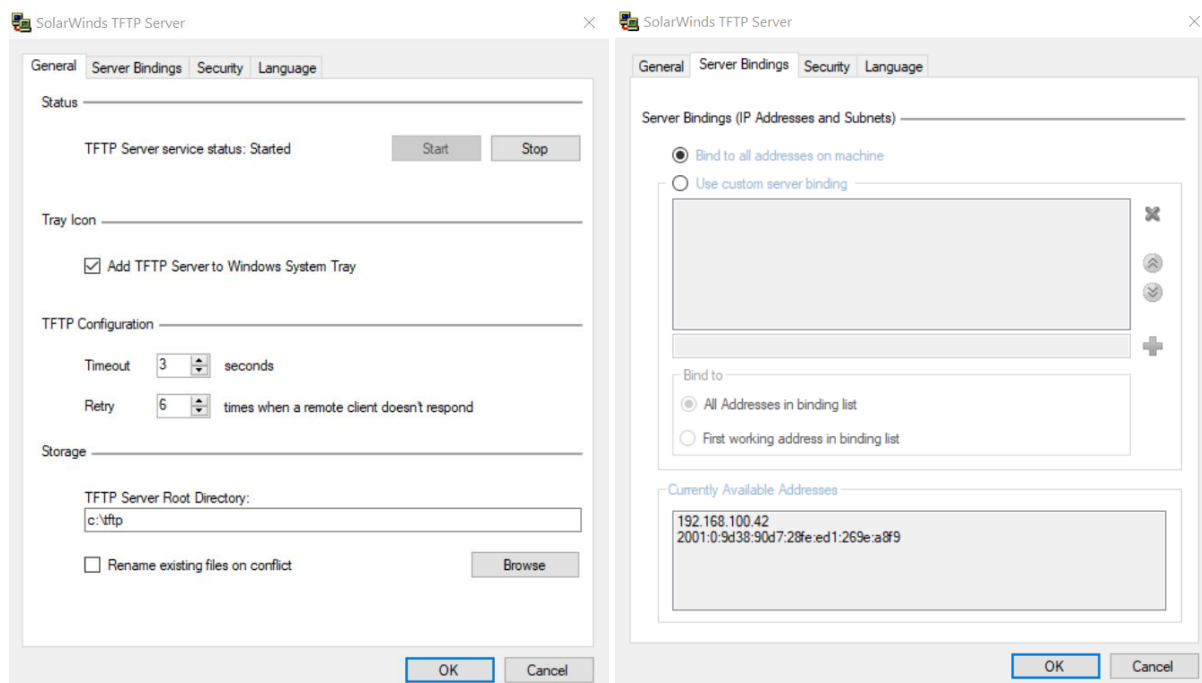
Nie ma zbyt wielu aplikacji wykorzystujących do transportu protokoły UDP. Posługują się nim przede wszystkim proste usługi, od których użytkownik wymaga dużej responsywności działania. Należą do nich np. usługa DNS i DHCP. Protokołu UDP używają również aplikacje transmitujące media, np. protokół VoIP. Do wygenerowania większej liczby datagramów UDP w celu ich analizy w programie Wireshark można wykorzystać aplikację TFTP (Trivial File Transfer Protocol). Aplikacja ma podobne cechy, co aplikacja FTP, jest jednak od niej dużo prostsza, bardziej prymitywna. Podobnie jak FTP aplikacja TFTP służy do przesyłania plików. System Windows nie

posiada jednak wbudowanego serwera TFTP. Należy posłużyć się dodatkową aplikacją. Prostą w konfiguracji i darmową aplikacją serwera TFTP jest aplikacja firmy SolarWinds.



Rysunek 7 - Aplikacja TFTP firmy SolarWinds.

Przed rozpoczęciem przysyłania plików Serwer TFTP należy skonfigurować do swoich celów. Serwer konfiguruje się w Menu File>Configure. Po wybraniu tej opcji otwiera się dodatkowe okno z opcjami konfiguracyjnymi. W oknie konfiguracyjnym występują cztery zakładki.

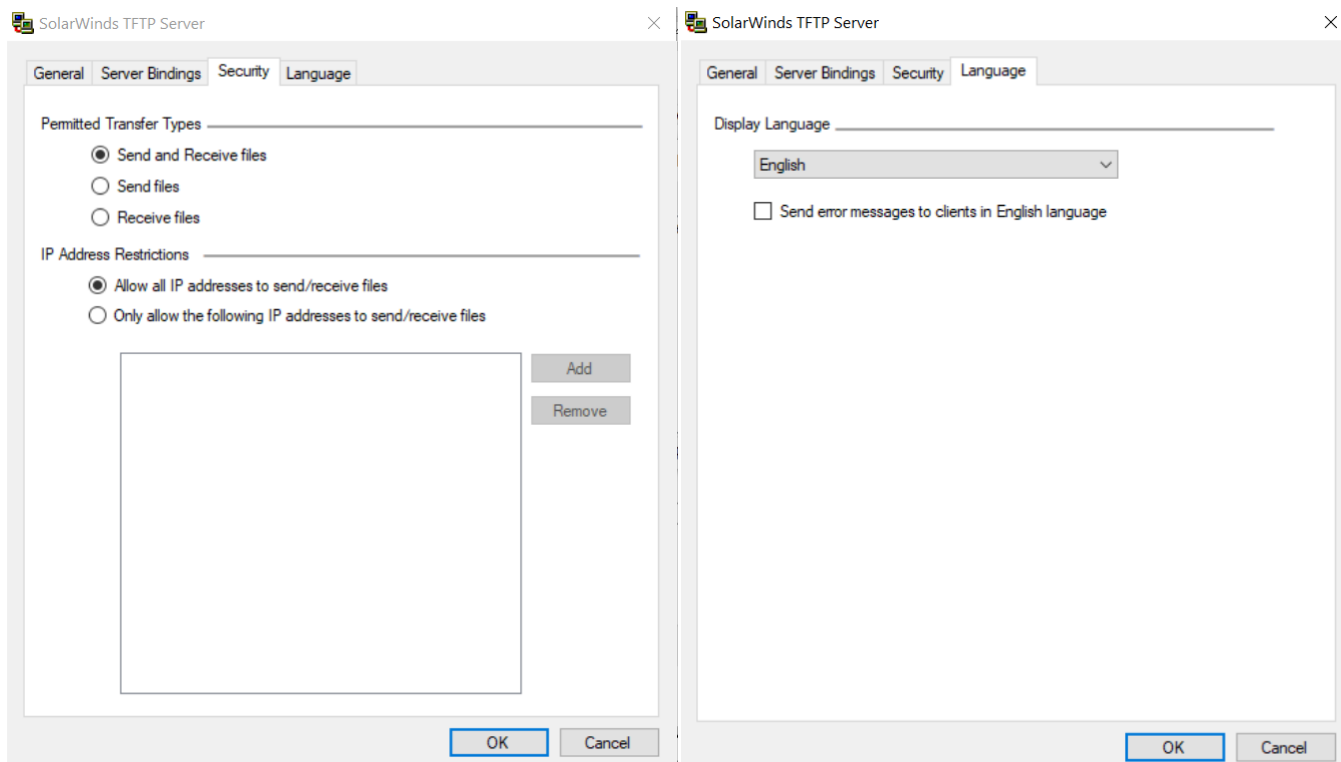


Rysunek 9 - Aplikacja TFTP firmy SolarWinds.
Konfiguracja katalogu głównego.

Rysunek 8 - Aplikacja TFTP firmy SolarWinds.
Konfiguracja używanych interfejsów.

W pierwszej zakładce (General) uruchamiamy lub zatrzymujemy serwer TFTP oraz konfigurujemy katalog główny, gdzie przechowywane są pliki. W drugiej (Server Bindings) konfigurujemy interfejsy sieciowe, których serwer może używać. W trzeciej zakładce konfigurujemy zabezpieczenia serwera przed nieautoryzowanym dostępem: opcje pozwolenia na kopiowanie lub wysyłanie plików oraz adresy IP, z których można połączyć się do serwera. Ostatnia zakładka zawiera wybór języka aplikacji.

Po skonfigurowaniu serwera, należy w katalogu roboczym TFTP umieścić plik do przesłania.



Rysunek 11 - Aplikacja TFTP firmy SolarWinds. Konfiguracja dozwolonego kierunku przesyłania danych oraz adresów IP klienta.

Rysunek 10 - Aplikacja TFTP firmy SolarWinds. Konfiguracja języka aplikacji.

Jako aplikację kliencką najprościej jest wykorzystać oprogramowanie wbudowane systemu Windows. W oknie Windows cmd uruchamiamy polecenie `tftp`. W odpowiedzi system wypisuje listę dostępnych parametrów. Program działa w trybie wsadowym. Wszystkie parametry należy podać w linii komend, zaraz za poleceniem `tftp`.

```

C:\tftp>tftp

Transfers files to and from a remote computer running the TFTP service.

TFTP [-i] host [GET | PUT] source [destination]

-i          Specifies binary image transfer mode (also called
             octet). In binary image mode the file is moved
             literally, byte by byte. Use this mode when
             transferring binary files.
host        Specifies the local or remote host.
GET         Transfers the file destination on the remote host to
             the file source on the local host.
PUT         Transfers the file source on the local host to
             the file destination on the remote host.
source      Specifies the file to transfer.
destination Specifies where to transfer the file.

C:\tftp>

```

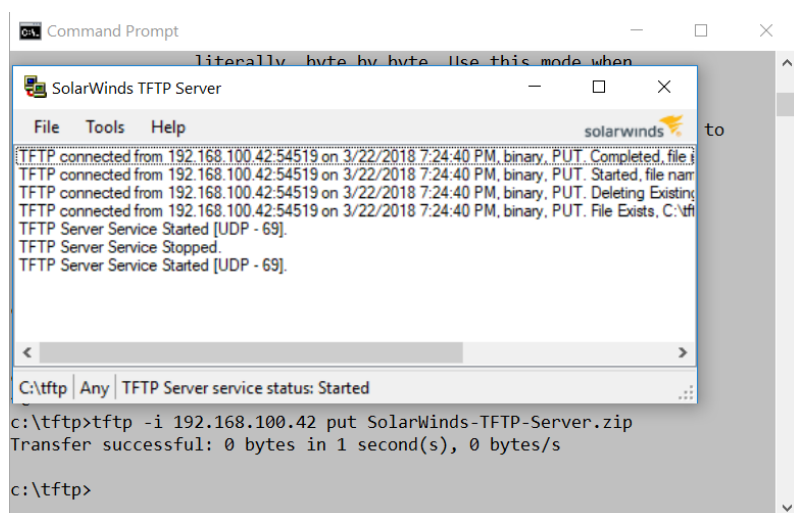
Figure 12 - Available options of TFTP MS Windows client.

Pierwsza opcja jest dosyć ważna. Należy jej używać zawsze, gdy przesyłamy plik binarny. W przeciwnym wypadku transfer może się zatrzymać już po pierwszej przesłanej ramce. Przykładowe polecenia wysyłania i pobierania pliku:

```

c:\tftp>tftp -i 192.168.0.12 put SolarWinds-TFTP-Server.zip
c:\tftp>tftp -i 192.168.0.12 get maly_plik.txt

```



Rysunek 13 - Wysłanie pliku na serwer TFTP. Komunikacja Windows cmd TFTP klient i serwer TFTP SolarWinds. Widoczne logi serwera.

Bibliography

Teaching materials of the Cisco Academy CCNA 1

Laboratory tasks

Transport Layer of the ISO-OSI model

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Basic information

Variables and symbols used in the instruction

- X - the variable X corresponds to the device number, as well as the laboratory group number.
- Y – universal variable correspond to device number of any other group.
K, L, M, N, ... - variables used interchangeably with the variables X and Y. Correspond to the devices and groups numbers.
Note: The variables K, L, M, N, ... are used when the order of the connected devices is important. For example, when configuration on the R_K router differs from the R_L router configuration. The variable L should correspond to the number of the current device (current laboratory group number), variable K indicates the “left” device and M the “right” device.
Variables X and Y are used when the order of the connected devices is not important. X means local device and Y means any remote device.
- A, B, C, ... - variables corresponding to the addresses of the networks created in the labs.
- [1.2.3.4] – the number of the CCNA lab corresponding to the current exercise.
- {Variable} - marks the required variable.
- [Variable] – marks an alternative variable.
- 2 points - task difficulty measure.
- COLORS - can be used to express some additional meaning:
 - # **red - means extremely important things and obligatory tasks;**
 - # **orange - means additional/alternative tasks;**
 - # **gray - means content that can be omitted in standard mode;**
 - # **green – means additional information, explanations;**
 - # **blue – means information to note to be checked by the instructor.**
- FONT
 - # **bold - highlighted information.**
 - # `Courier New - configuration commands.`

IP addressing and device names used in the exercise

PCs

Name:	PC_X1
Net:	LAN_X (Cisco)
IP PC_L1:	192.168.X.X1/24
	fc00:X::X1/64
	fe80::X1
Gateway:	192.168.X.X0

fc00:X::X0/64

Name: PC_X2
 Net: LAN_X (Cisco)
 IP PC_L2: 192.168.X.X2/24
 fc00:X::X2/64
 fe80::X2
 Gateway: 192.168.X.X0
 fc00:X::X0/64

Switch

Name: S_X
 Net: LAN_X
 Interface: VLAN 1
 IP: 192.168.X.X9/24
 fc00:X::X9/64
 fe80::X9

Native VLAN: VLAN 1
 Default gateway: 192.168.X.X0
 fc00:X::X0

Router

Name: R_X
 Net: LAN_X
 Interface: Fa0/0 (Fa0/2/0 or another connected)
 IP: 192.168.X.X0/24
 fc00:X::X0/64
 fe80::X0

X, L – the number of the configured device (X = L).

K – the previous device number.

M – next device number.

Y – any remote device number.

Basic topology of a single laboratory group

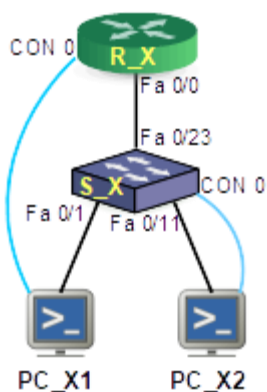


Figure 14 - Basic topology of a single laboratory group.

The following connections should be configured in the baseline topology:

PC_X1 Cisco NIC port connected with straight cable to S_X port Fa0/1.
 PC_X1 COM port 1 connected with console cable to R_X CON port 0.
 PC_X2 Cisco port NIC connected with straight cable to S_X port Fa0/11.
 PC_X2 COM port 1 connected by a console cable to S_X CON CON port 0.
 R_X Fa 0/0 port connected by a straight cable to S_X port Fa0/23.

Lab topology

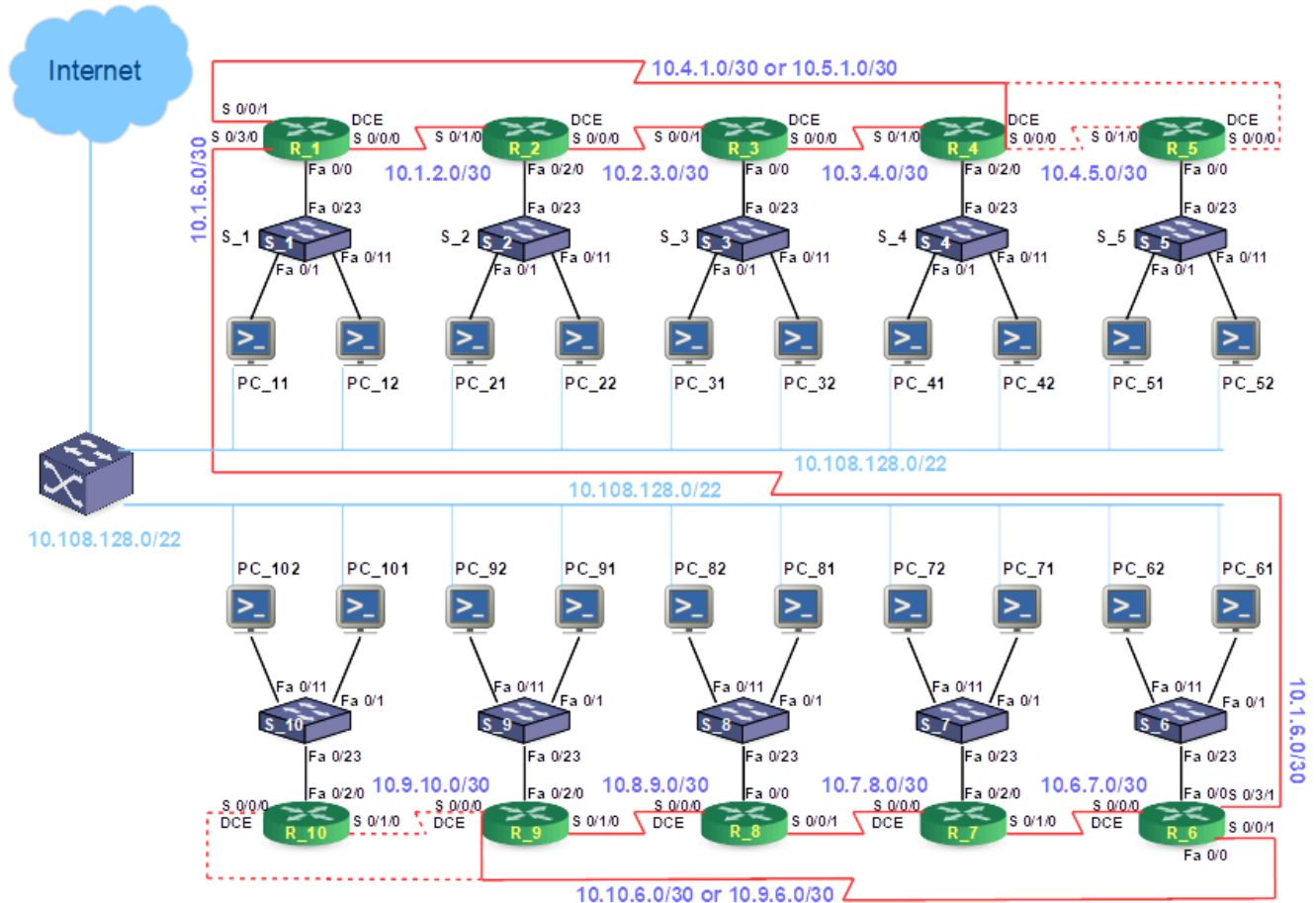


Figure 15 - Laboratory topology with double cabling.

Base PC configuration

Configure the computers using the IP addresses given in the instruction. Set the static IP address on the internal NIC ("Cisco") that leads to the lab set. Set the dynamic IP address (DHCP) on the external NIC ("Internet"). Check the correctness of the configuration.

```
c:\>ipconfig
```

References to the CCNA 1 v. 6.0 labs

- [9.2.1.6] - Using Wireshark to Observe the TCP 3-Way Handshake
- [9.2.3.5] - Using Wireshark to Examine a UDP DNS Capture
- [9.2.4.3] - Using Wireshark to Examine FTP and TFTP Captures

References to the CCNA 1 v. 6.0 Packet Tracer scripts

- [9.3.1.2] - TCP and UDP Communications

Note: Each task can be performed in two-person groups. Each of the tasks can be performed jointly or entire work can be divided into individual goals between group members.

Task 1 – Analysis of the TCP protocol in Wireshark

Points: 1

Objectives

- Analysis of TCP protocol operation.
- Determining the phases of protocol operation:
 - The phase of establishing a connection (Three Way Handshaking);
 - Data transfer phase;
 - The termination phase of the TCP connection.

Practical task

The administrator in the company got the task of analyzing TCP connections in order to diagnose existing problems. Particular attention should be paid to the size of the TCP window configured for the transmission of large files.

Network topology

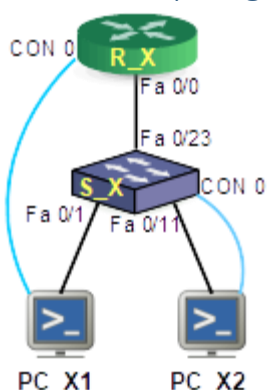


Figure 16 - Base topology of a single laboratory group.

The following connections should be configured in the baseline topology:

- PC_X1 Cisco NIC port connected with straight cable to S_X port Fa0/1.
- PC_X1 COM port 1 connected with console cable to R_X CON port 0.
- PC_X2 Cisco port NIC connected with straight cable to S_X port Fa0 / 11.
- PC_X2 COM port 1 connected by a console cable to S_X CON CON port 0.
- R_X Fa 0/0 port connected by a straight cable to S_X port Fa0 / 23.

The exercise should be made in basic lab topology. To connect group's computers use the Cisco or Internet switch.

Exercise scenario

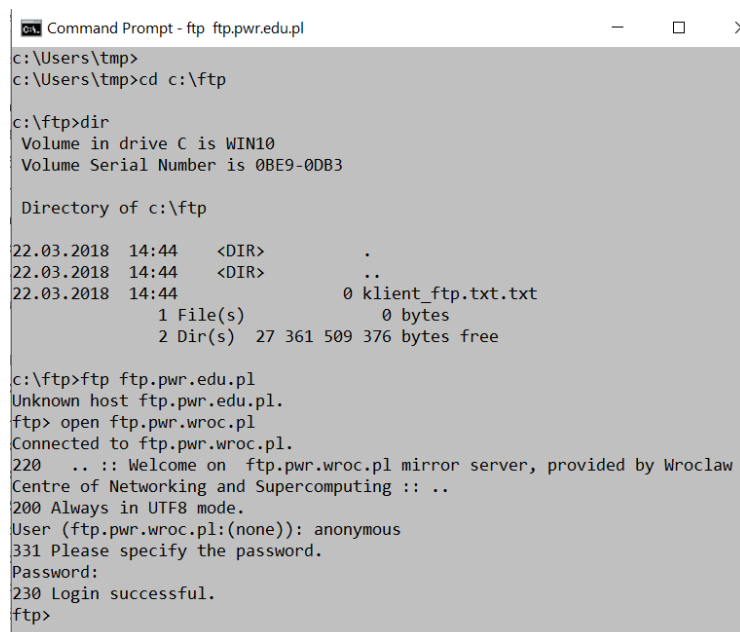
1. Create a text document in the user's home directory (eg: C:\cisco\users\user.name) and [save the course of exercise](#) in it.
2. Configure the statics IP addresses on internal interfaces (Cisco) under MS Windows system. Use the addresses given in the instruction. Configure dynamic IP address (DHCP) on external NIC ("Internet").

Check the correctness of configuration using *ipconfig* command.

```
c:\>ipconfig
```

3. Prepare the computers for FTP transfer.
4. Configure the PC_X1 computer as the FTP client
 - a. In the *c:\ftp* folder, create the text file named *client_ftp.txt*. Save your name inside the file.
 - b. Open *cmd* terminal and change the current directory to *c:\ftp*.
 - c. Check PC_X2 IP address.
 - d. Test the connectivity with *ping* command.
5. Configure the PC_X2 computer as the FTP server. Use the FTP server build into MS Windows Server. The root directory of the MS Windows FTP server is configured as *c:\ftp* folder.
 - a. Inside the *c:\ftp* folder create the text file named *server_ftp.txt*. Save your name inside the file.
6. Start the Wireshark on adequate interface.
7. Transfer the created text files between your computers using the FTP protocol.

Make the FTP connection from PC_X1 to PC_X2 computer. From the PC_X1 command line window start the built in FTP client.



```

Command Prompt - ftp.pwr.edu.pl
c:\Users\tmp>
c:\Users\tmp>cd c:\ftp

c:\ftp>dir
Volume in drive C is WIN10
Volume Serial Number is 0BE9-0DB3

Directory of c:\ftp

22.03.2018  14:44    <DIR>        .
22.03.2018  14:44    <DIR>        ..
22.03.2018  14:44                0 klient_ftp.txt.txt
               1 File(s)                0 bytes
               2 Dir(s)  27 361 509 376 bytes free

c:\ftp>ftp ftp.pwr.edu.pl
Unknown host ftp.pwr.edu.pl.
ftp> open ftp.pwr.wroc.pl
Connected to ftp.pwr.wroc.pl.
220  .. :: Welcome on  ftp.pwr.wroc.pl mirror server, provided by Wroclaw
Centre of Networking and Supercomputing :: ..
200 Always in UTF8 mode.
User (ftp.pwr.wroc.pl:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp>
  
```

Figure 17- Establishing an ftp connection via the build in ftp client from the Windows cmd console.

Follow the scenario of actions:

- a. Login with Administrator or anonymous account.
- b. List the remote directory content.

```
ftp>dir
```

- c. Download the text file from the FTP server: *server_ftp.txt*.

```
ftp>?
```

```
ftp>ascii
```

```
ftp>hash
```

```
ftp>get server_ftp.txt
```

- d. Sent the file to server: *client_ftp.txt*.

```
ftp>put client_ftp.txt
```

- e. Finish the FTP connection.

ftp>bye

8. Stop the Wireshark. [Save](#) the results into the user folder.
9. Inside the Wireshark listing find and [note the numbers of the following frames](#):
 - a. Frames corresponding for [establishing the TCP connection](#).
 - b. Frames with FTP login [name and password](#).
 - c. Frames with [directory listing](#) – should contain list of FTP server files in TCP data part.
 - d. Frames containing [copied file](#) – should contain the user name in TCP data part.
 - e. Frames corresponding to [connection end](#).
10. Expand the TCP abbreviation.
11. You can ask the instructor to check your current solutions or to continue solving the next task.

Task 2 – Analysis of the UDP protocol

Points: 1

Objectives

→ Analysis of the UDP protocol operation.

Practical task

The administrator in the company got the task of analyzing UDP connections in order to diagnose existing problems.

Network topology

The exercise should be made in basic lab topology. To connect group's computers, use the Cisco or Internet switch.

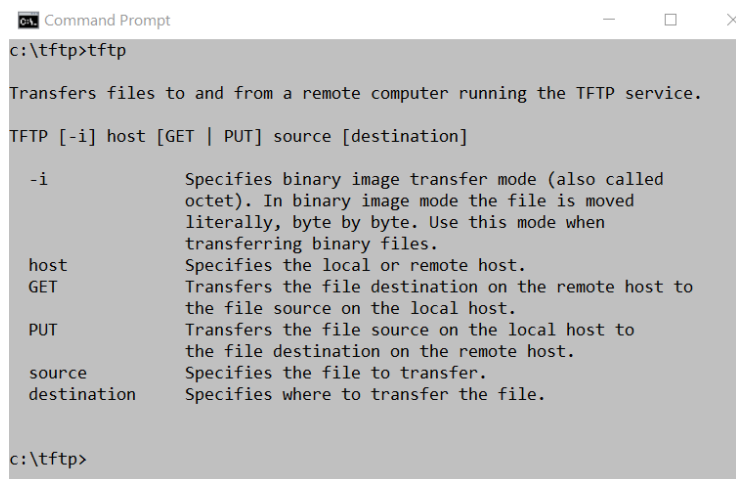
Exercise scenario

1. Create a text document in the user's home directory (eg: C:\cisco\users\user.name) and [save the course of exercise in it](#).
2. Configure the statics IP addresses on internal interfaces (Cisco) under MS Windows system. Use the addresses given in the instruction. Configure dynamic IP address (DHCP) on external NIC ("Internet"). Check the correctness of configuration using *ipconfig* command.

```
c:\>ipconfig
```
3. Prepare the computers for TFTP transfer.
4. Configure the PC_X2 computer as the TFTP client. Use the built into MS Windows Server TFTP client or use alternative software.
 - a. Inside the *c:\tftp* folder create the text file named *client_tftp.txt*. Save your name inside the file.
 - b. Open *cmd* terminal and change the current directory to *c:\tftp*.
 - c. Check PC_X1 IP address.
 - d. Test the connectivity PC_X1 – PC_X2 with *ping* command.
5. Configure the PC_X1 computer as the TFTP server. Use the TFTP SolarWinds server application or an alternative software. The root directory of the TFTP server should be configured as *c:\tftp*.

- a. Inside the `c:\tftp` folder create the text file named `server_tftp.txt`. Save your name inside the file.
 - b. Start the TFTP server. Check correctness of its configuration.
6. Start the Wireshark on adequate interface.
 7. Make the text file transfer using TFTP protocol.

Start the TFTP connection from the PC_X2 to PC_X1 computer. From the PC_X2 command line window start the built into MS Windows TFTP client.



```

c:\tftp>tftp

Transfers files to and from a remote computer running the TFTP service.

TFTP [-i] host [GET | PUT] source [destination]

-i           Specifies binary image transfer mode (also called
             octet). In binary image mode the file is moved
             literally, byte by byte. Use this mode when
             transferring binary files.
host         Specifies the local or remote host.
GET          Transfers the file destination on the remote host to
             the file source on the local host.
PUT          Transfers the file source on the local host to
             the file destination on the remote host.
source       Specifies the file to transfer.
destination  Specifies where to transfer the file.

c:\tftp>

```

Figure 18 - Available options of TFTP MS Windows client.

Follow the scenario of actions:

- a. Start the TFTP connection ?
 - b. List the remote directory ?
 - c. Download the file from server: `server_tftp.txt`.
 - d. Sent the file to server: `client_tftp.txt`.
 - e. Finish the TFTP connection ?
8. Stop the Wireshark. **Save** the results into the user folder.
 9. Inside the Wireshark listing find and **note the numbers of the following frames**:
 - a. Frames corresponding for **establishing the UDP connection** ?
 - b. Frames with TFTP **login name and password** ?
 - c. Frames with **directory listing** ?
 - d. Frames containing **copied file** – should contain the user name in UDP data part.
 - e. Frames corresponding to **connection end** ?
 10. Expand the UDP abbreviation.
 11. You can ask the instructor to check your current solutions or to continue solving the next task.
Remember to **save** your answers.

Task 3 – Comparison of TCP and UDP protocols

Points: 2

Objectives

- Checking the transfer of large amounts of data via the TCP protocol.

- Use of the FTP protocol to transfer a large file.
- Checking the transfer of large amounts of data via the UDP protocol.
- Use TFTP protocol to send a large file.

Practical task

The company administrator decides to analyze and optimize the transport protocols (TCP and UDP). He would like to check the TCP window size.

Network topology

The exercise should be made in basic lab topology. To connect group's computers, use the Cisco or Internet switch.

Exercise scenario

1. Create a text document in the user's home directory and [save the course of exercise](#) in it.
2. Configure the static IP addresses on internal interfaces (Cisco) under MS Windows system. Use the addresses given in the instruction. Configure dynamic IP address (DHCP) on external NIC ("Internet"). Check the correctness of configuration using *ipconfig* command.

```
c:\>ipconfig
```

TCP transfer

3. Prepare the computers for FTP (TCP) transfer.
4. Copy the binary file with the size of several dozen MB to the FTP root directory *c:\ftp*. This can be an IOS image of the router found in the *c:\cisco\ios* directory
5. Start the Wireshark program on chosen interface.
6. Make a large file copy between two computers.
7. Stop the FTP session.
8. Stop the Wireshark program. [Save](#) the captured frames into the user folder.
9. Analyze the captured frames from Wireshark. Use the appropriate filter to facilitate your work e.g.:

```
tcp and ip.addr=={remote PC address}
```

[Note the answers](#) for the following questions:

- a. Which [default ports](#) (find two) are used by the FTP protocol (number <1024)
- b. Maximum data frames sent without any acknowledgement.
- c. Find the data frames (fragment of file) and the corresponding ACK confirmation for this data.
- d. What is the size of the TCP window set by the FTP server and the FTP client?
- e. Using TCP Window calculate the maximum number of frames the FTP server could send without confirmation?

UDP transfer

10. Prepare the computers for TFTP (UDP) transfer.
11. Copy the binary file with the size of several dozen MB to the TFTP root directory *c:\tftp*. This can be an IOS image of the router found in the *c:\cisco\ios* directory
12. Start the Wireshark program on chosen interface.
13. Make a large file copy between two computers.
14. Stop the Wireshark program. [Save](#) the captured frames into the user folder.
15. Analyze the captured frames from Wireshark. Use the appropriate filter to facilitate your work e.g.:

```
udp and ip.addr=={remote PC address}
```

Note the answers for the following questions:

- a. Which default ports are used by the TFTP protocol (number <1024)?
 - b. Maximum number of data frames sent without acknowledgement?
 - c. Find the data frames (fragment of file) and the corresponding ACK confirmation for this data.
 - d. Calculate the maximum number of frames the TFTP server could send without confirmation ?
 - e. How does the TFTP protocol ensure error-free data transmission (e.g. for IOS image)?
16. Report the results stored in the text file to the instructor. Alternatively, save the answers and proceed to the next task.

Additional Task 4 – Comparison of TCP and UDP protocols in Linux

Points: 2

Objectives

- Analyzing the TCP and UDP protocol in Linux.
- Comparison of implementation and operation of transport protocols under Windows and Linux.

Practical task

The administrator got the task of optimizing TCP and UDP connectivity between different operating systems. In practice, it turned out that the performance of programs copying data differs in different systems.

Network topology

The exercise should be made in basic lab topology. To connect group's computers, use the Cisco or Internet switch.

Exercise scenario

1. Complete the previous task in the Linux operating system.
2. Create a text document in the user's home directory and save the course of exercise in it.
3. Configure PCs to work in the laboratory network. Use the IP addresses provided in the instructions. Be sure to set the dynamic DHCP configuration on external interface.

TCP transfer

4. Prepare the computers for TCP file transfer in Linux.
 - a. Reset the computer and select the Linux Debian operating system at startup.
 - b. Log in using the data provided by the teacher. For example


```
login:      stud
password:   stud
login:      root
password:   P@ssw0rd
```
 - c. Check if the ftp server is installed in the system. If not, install and configure the ftp server. Use the following commands:


```
aptitude install -R proftpd
```
 - d. Check if the ftp client is installed on the system. If not, install and configure the application.

5. Check if the Wireshark program is installed on the system. If it is missing, install it.

```
aptitude install -R wireshark
```

6. Start the Wireshark program on chosen interface.
7. Make a large file copy between computers on the network.
8. Analyze the TCP protocol operation. Compare the results with those obtained under Windows. [Make a note](#) of the observations and results in the text file. [Answer the same questions](#) as in the previous task.

UDP transfer

9. Configure computers to transfer files using the UDP protocol in Linux.
 - a. Check if the tftp client and server are installed in the system. If not, install and configure these applications. Use, for example, the commands:


```
aptitude install -R tftp tftpd
or
aptitude install -R tftp atftpd (advanced version)
```
10. Make a large file copy between computers in the network.
11. Analyze the UDP protocol operation. Compare the results with those obtained under Windows. [Make a note](#) of the observations and results in the text file. [Answer the same questions](#) as in the previous task.
12. Report the results stored in the text file to the instructor. Alternatively, [save](#) the answers and proceed to the next task.

Additional Task 5 – Backup of IOS and configuration on Cisco devices.

Use of UDP protocol

Points: 1

Objectives

- Checking the way of archiving data on Cisco devices.
- Checking the reliability of data archiving.

Practical task

The administrator got the task of checking TFTP communication between Cisco devices and the backup server. Some images are not properly archived on the server.

Network topology

The exercise should be made in basic lab topology. To connect group's computers, use the Cisco switch.

Exercise scenario

1. Check the PC's IP configuration on internal interfaces leading to lab set.
2. Configure the computers for TFTP file transfer.
3. Start the TFTP servers (e.g. SolarWinds) on both computers.
4. Configure IP address on the switch S_X VLAN1 interface. It is needed to enable switch to IP network communication.

```
Switch>enable
Switch#configure terminal
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.X.X9 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#end
```

5. Configure IP address on router R_X LAN interface.

```
Router>enable
Router#configure terminal
Router(config)#interface FastEthernet 0/0 (or another connected)
Router(config-if)#ip address 192.168.X.X0 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#end
```

6. Make the connectivity test between router, switch and computers.

7. In the case of problems correct the configuration.

8. Start the Wireshark on internal interfaces of both computers.

9. Copy the configuration and then IOS of the R_X router to TFTP server on PC_X1.

```
Router#copy running-config tftp
... specify the TFTP IP address and target file name
Router#copy flash: tftp:
```

10. Copy the configuration and then IOS of the R_X router to TFTP server on PC_X2.

```
Switch#copy running-config tftp
... specify the TFTP IP address and target file name
Switch#copy flash: tftp:
```

11. Stop the Wireshark program. [Save](#) the captured frames into the user folder.

12. Analyze the captured frames. Use the appropriate filter e.g.:

```
udp and ip.addr== {TFTP server address}
```

[Note the answers](#) for the following questions:

- Which [default ports](#) are used by the TFTP protocol (number <1024)?
 - How many [data frames](#) are [sent without acknowledgement](#)?
 - Find the data frames (fragment of file) and the corresponding ACK confirmation for this data.
 - What maximum number of frames the TFTP server could send without confirmation?
 - How does the TFTP protocol ensure error-free data transmission (e.g. for IOS image)?
13. Report the results stored in the text file to the instructor. Alternatively, [save](#) the answers and proceed to the next task.

Laboratory tasks in a virtual environment. Packet Tracer simulator

Transport Layer of the ISO-OSI model

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

The Packet Tracer environment allows analysis of the TCP and UDP protocol. However, you cannot view the transmitted frames in Wireshark.