**Obraz zawierający tekst, clipart, grafika wektorowa

Opis wygenerowany automatycznie**

**Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,**

**INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

**KATEDRA METROLOGII I ELEKTRONIKI**

Technologie telekomunikacyjne  
**Projekt Ethernet**

Autorzy: Krzysztof Cisło i Jakub Działowy

Kierunek studiów: Mikroelektronika w Technice i Medycynie

Kraków, 2021

Spis treści

[1. ARP 3](#_Toc90039252)

[1.1. Ćwiczenie: ARP scanner 3](#_Toc90039253)

[2. ICMP 5](#_Toc90039254)

[2.1. Ćwiczenie: Ping program 5](#_Toc90039255)

[3. UDP 7](#_Toc90039256)

[3.1. Ćwiczenie: Odebranie aktualnej daty i czasu 7](#_Toc90039257)

[4. TCP 8](#_Toc90039258)

[4.1. Ćwiczenie: wysłanie strony HTML z wykorzystaniem http 8](#_Toc90039259)

[4.2. Ćwiczenie: odebranie strony HTML 9](#_Toc90039260)

# ARP

ARP (Address Resolution Protocol), to protokół sieciowy umożliwiający mapowanie logicznych adresów warstwy sieciowej na fizyczne adresy warstwy łącza danych. Wykorzystywany jest np. w ramach większych systemów TCP/IP do weryfikacji adresów MAC przed wysłaniem dalszych wiadomości lub do identyfikacji potencjalnych konfliktów adresów IP, ponieważ różne węzły o identycznych adresach IP będą miały różne adresy MAC.

## Ćwiczenie: ARP scanner

Celem ćwiczenia jest stworzenie prostego skanera ARP, który sprawdza dostępne adresy MAC dla zakresu adresów IP od 192.168.0.0 do 192.168.0.255. Program skanuje adresy IP w danej sieci poprzez wysłanie zapytań pod kolejne adresy z podanego zakresu, a gdy uzyska odpowiedź z adresem MAC, to wyświetla w konsoli IP i MAC urządzenia. Kod programu przedstawiono na listingu 1. Przy wywołaniu funkcji skorzystano z maski /24, która pozwala pokryć zadany zakres adresów z polecenia do zadania.

Obraz zawierający tekst

Opis wygenerowany automatycznie

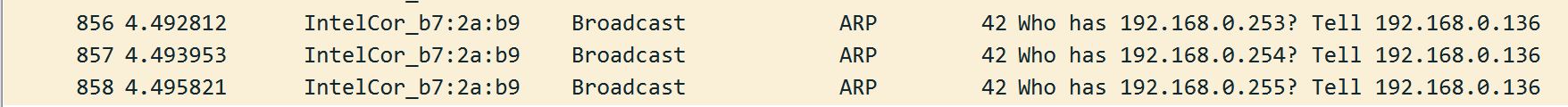
Listing 1. Kod programu wyświetlającego IP i MAC urządzeń podłączonych do sieci.

Za pomocą programu Wireshark obserwowano wysyłane i odbierane pakiety protokołu ARP, widoczne na rysunkach 1. i 2. Zapytania są wysyłane pod kolejne adresy IP, a gdy dany adres jest dostępny, to zgodnie z oczekiwaniami odsyła adres MAC. Wydruk programu w konsoli przedstawiono na rysunku 4.

Obraz zawierający stół

Opis wygenerowany automatycznie

Rysunek 1. Początkowy fragment rozwiązania w Wireshark’u.



Rysunek 2. Końcowy fragment rozwiązania w Wireshark’u.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 3. Rezultat uruchomionego programu ARP.

# ICMP

Protokół ICMP (Internet Control Message Protocol) jest powszechnie używany do sprawdzania połączeń z innymi komputerami. Potocznie nazywa się „pingowaniem”, ponieważ działa w taki sam sposób, jak „ping” sonaru wysyłany przez okręty podwodne. Jeśli trafi w cel, powróci echo, które można wychwycić i zbadać.

## Ćwiczenie: Ping program

Celem ćwiczenia jest program, który wysyła polecenie ping, czyli żądanie ICMP na adres IP. W przypadku otrzymania odpowiedzi wyświetla adres IP i powiązany adres MAC w konsoli. W programie przedstawionym na listingu 2. wykorzystano funkcję z wcześniejszego ćwiczenia do uzyskania adresu MAC poprzez protokół ARP. Dwa adresy IP są dostępne, natomiast jeden nie jest dostępny w celu sprawdzenia poprawności działania programu w obydwu przypadkach.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Listing 2. Kod programu wysyłającego ping i wyświetlającego IP oraz MAC po odebraniu odpowiedzi.

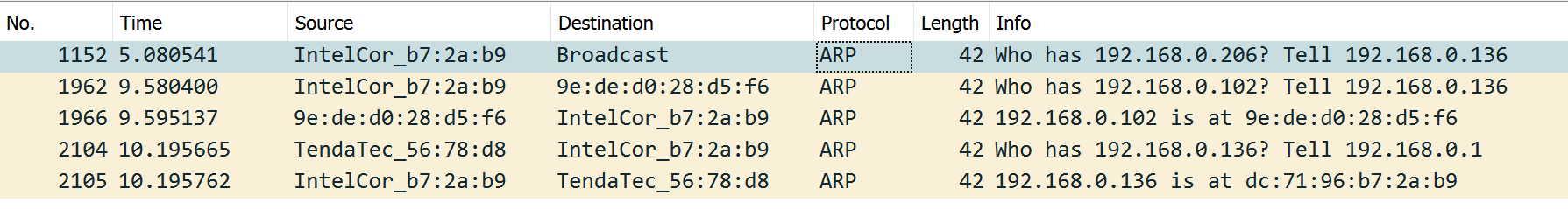
Jak widać na rysunku 4. od jednego z adresów IP nie uzyskano odpowiedzi. Natomiast dwa następne przesłały odpowiedź, a ich adresy IP i MAC zostały wypisane w konsoli.

Obraz zawierający tekst

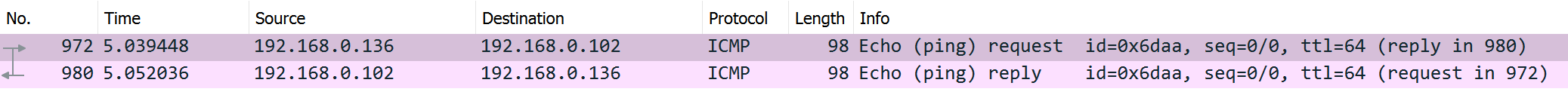
Opis wygenerowany automatycznie

Rysunek 4. Rezultat uruchomionego ping programu.

Wykorzystana w programie funkcja multiping w pierwszej kolejności z wykorzystaniem protokołu ARP sprawdza czy dane adresy IP są dostępne, co widać na rysunku 5. Następnie wysyła ping do tych dostępnych (Rysunek 6.).



Rysunek 5. Wynik w Wireshark’u po sprawdzeniu przez metodę multiping czy dany adres IP odpowiada.



Rysunek 6. Wynik wysłania i odebrania ping w Wireshark’u.

# UDP

UDP (User Datagram Protocol), to metoda wysyłania danych bezpośrednio do określonego socketu. UDP może być używany jako protokół komunikacji bezpośredniej, w którym wysyła się wiadomości do określonej aplikacji w określonym systemie. W UDP wysyła się sygnał do konkretnego portu i jeśli wszystko poszło dobrze, a aplikacja monitorująca port jest skonfigurowana tak, aby odpowiadać na przychodzące wiadomości, to port wygeneruje odpowiedź, która zostanie otrzymana, a w przeciwnym wypadku nic nie zostanie odebrane. Dzięki temu UDP jest przydatny w sytuacji wysyłania niestandardowych komunikatów danych bezpośrednio z jednego systemu do drugiego.

## Ćwiczenie: Odebranie aktualnej daty i czasu

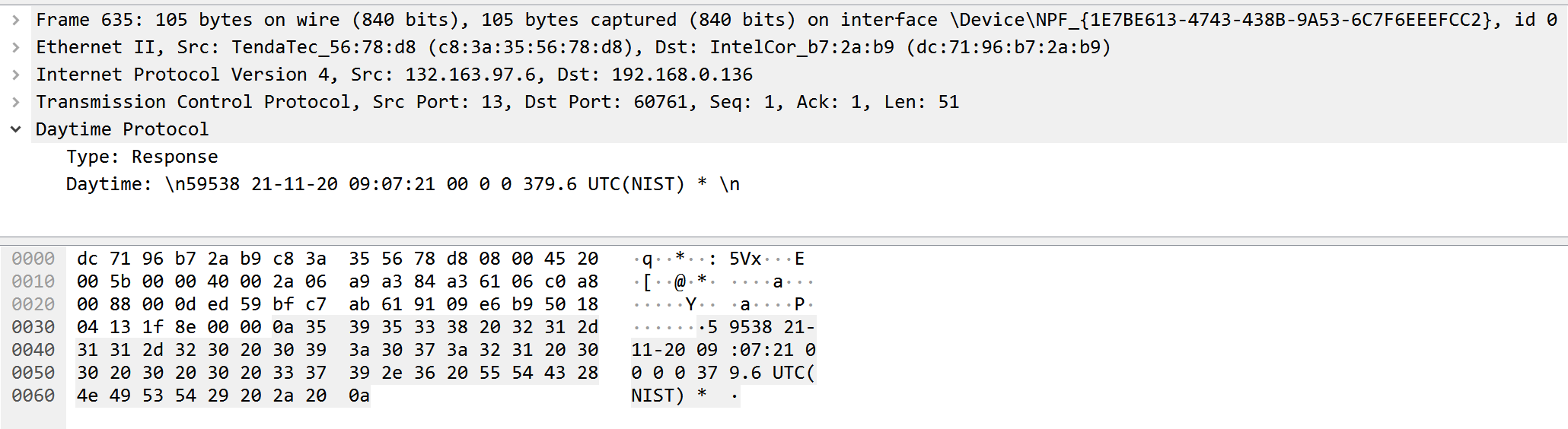
Celem ćwiczenia jest pobranie czasu i daty z socketu 13. na docelowym komputerze. Socket 13 odpowie na komunikat UDP, zwracając godzinę i datę w postaci ciągu ASCII. Treść programu przedstawiono na listingu 3.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Listing 3. Kod programu do odbierania daty i godziny z portu 13.

Wyniki działania programu przedstawiono na rysunkach 7. i 8. Odebrana wiadomość z socketu 13. jest ciągiem znaków zawierających datę i czas. Ten ciąg znaków wypisano w konsoli (Rysunek 8).



Rysunek 7. Wyniki działania programu w Wireshark’u.



Rysunek 8. Rezultat uruchomionego programu UDP widoczny w terminalu.

# TCP

TODO:

## Ćwiczenie: Wysłanie strony HTML z wykorzystaniem http

Celem ćwiczenia jest stworzenie strony internetowej i wyświetlenie jej w przeglądarce. Kod programu przedstawiono na listingu 4.



Listing 4. Kod programu wysyłający stronę HTML poprzez HTTP.

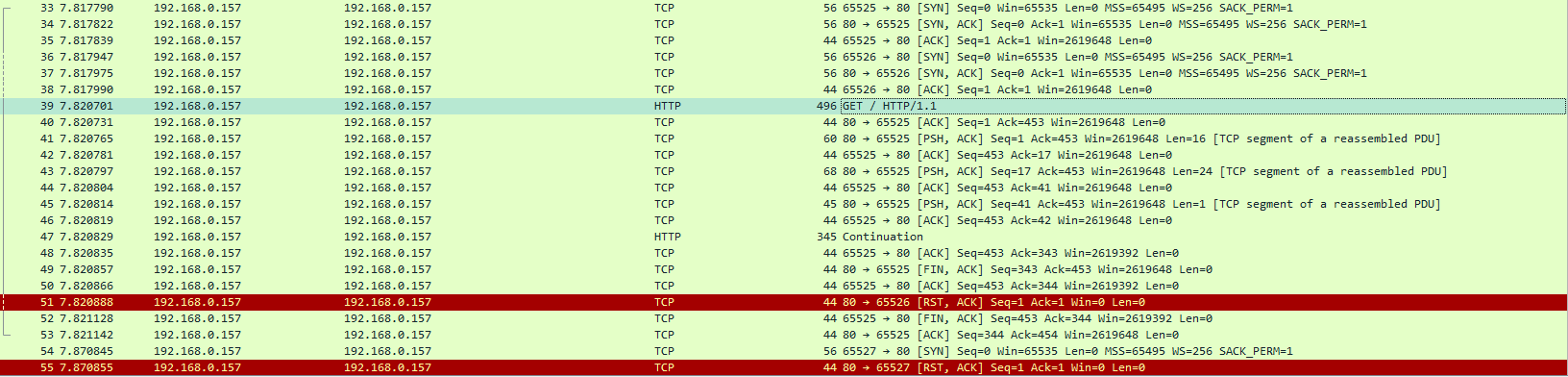
Działanie programu sprawdzono przy użyciu przeglądarki internetowej Google Chrome. Wpisano odpowiedni adres lokalny, w tym przypadku: 192.168.0.10, a następnie przeglądarka załadowała stronę, widoczną na rysunku 9.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 9. Strona załadowana przez przeglądarkę.

Ponadto, przechwycono pakiety przy użyciu programu Wireshark, widoczne na rysunku 10.



Rysunek 10. Przechwycone pakiety protokołu HTTP.

Na rysunku 11. przedstawiono pakiet zawierający dane strony HTML.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 11. Pakiet zawierający dane strony HTML.

## Ćwiczenie: Odebranie strony HTML

Celem ćwiczenia jest odebranie strony internetowej i wyświetlenie jej w konsoli. Kod programu przedstawiono na listingu 5.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Listing 5. Kod programu odbierający stronę HTML poprzez HTTP.

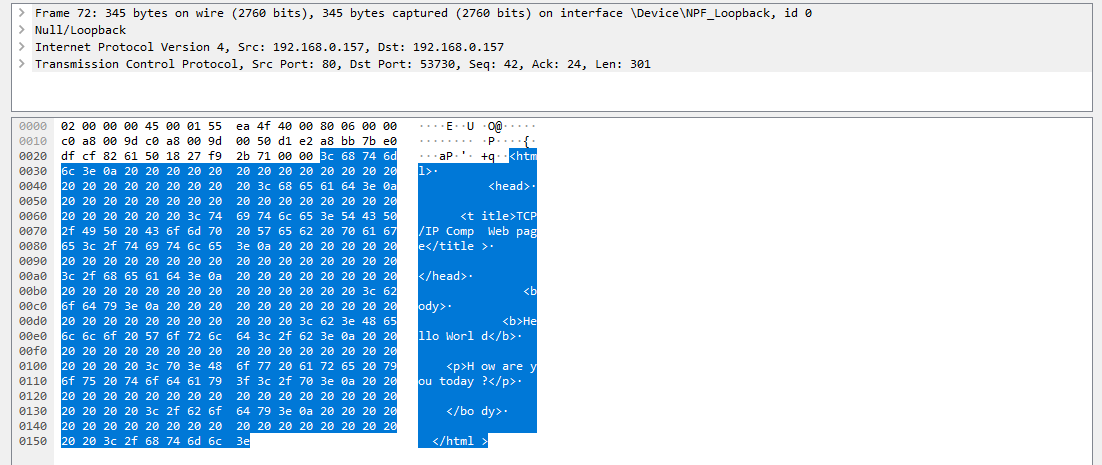
Działanie programu sprawdzono przy użyciu dwóch terminali. Jeden służył jako serwer i czekał na zapytanie. Natomiast drugi działał jako klient (analogicznie do przeglądarki internetowej). Klient po odebraniu danych wypisał zawartość przesłanej strony, co przedstawiono na rysunku 12.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 12. Odebrana strona HTML wyświetlona w konsoli.

Podczas tego testu przechwycono dane przy użyciu programu Wireshark (Rysunek 13).



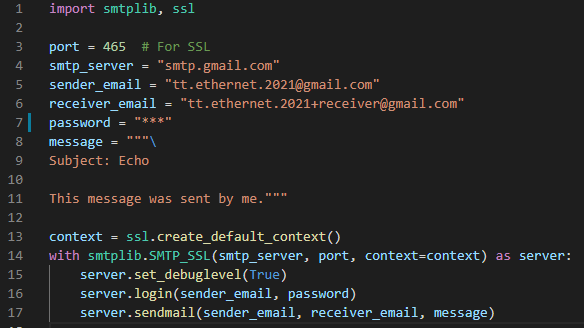
Rysunek 13. Przechwycone dane strony HTML podczas komunikacji serwera i klienta.

# SMTP

TODO:

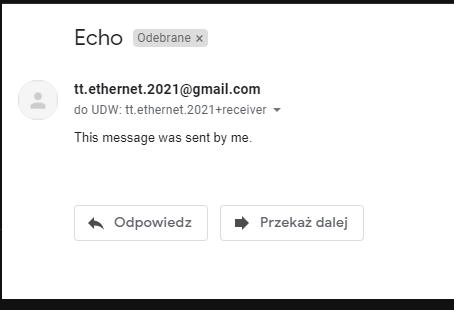
## Ćwiczenie: Wysłanie maila.

Celem ćwiczenia było wysłanie prostego maila przy użyciu SMTP. Kod programu przedstawiono na listingu 6. Wykorzystano bibliotekę smtplib oraz szyfrowanie SSL. Na potrzeby tego ćwiczenia utworzono również skrzynkę mailową Gmail.



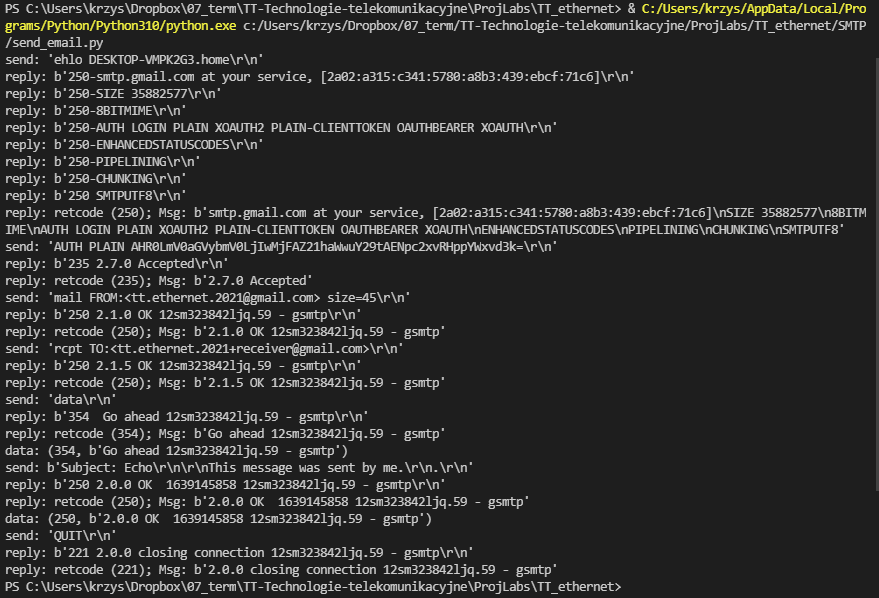
Listing 6. Kod programu wysyłającego maila poprzez SMTP.

Działanie programu sprawdzono wprost w skrzynce mailowej. Po włączeniu programu otrzymano maila widocznego na rysunku 14.



Rysunek 14. Otrzymany testowy mail.

Przechwycenie danych w Wireshark’u było niemożliwe ze względu na użycie serwera smtp@gmail.com, dlatego wykorzystano tryb „debug” obiektu SMTP\_SSL. Zaobserwowane dane przedstawiono na rysunku 15. Wyraźnie widać interaktywny proces i charakterystyczne: „mail FROM”, „rcpt TO”, „data” oraz „QUIT”.



Rysunek 15. Wysyłane i odbierane dane protokołu SMTP.

## Ćwiczenie: Wysłanie maila z zawartością HTML.

Celem ćwiczenia: TODO

Obraz zawierający tekst

Opis wygenerowany automatycznie

Listing 7. Część pierwsza kodu programu wysyłającego maila z zawartością HTML.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Listing 8. Część druga kodu programu wysyłającego maila z zawartością HTML.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 16. Odebrany testowy mail z zawartością HTML.