

Warsztaty z Sieci komputerowych

Lista 4

Zadanie 1. Wszystkie komputery są wpięte do jednego przełącznika kartami `eth0`. Rozpocznij pracę od polecenia `sudo netmode lab`.

Uruchom Wiresharka i włącz w nim obserwację interfejsu `eth0`. Jeśli otrzymujesz dużo komunikatów od przełącznika, możesz je odfiltrować klikając prawym przyciskiem myszy odpowiedni komunikat, wybierając z kontekstowego menu *prepare as filter — not selected* a następnie przycisk *apply*. W konsoli wydaj polecenie

```
#> dhclient eth0
```

Jakie komunikaty zostają wymienione pomiędzy Twoim komputerem a serwerem DHCP? Zauważ, że DHCP posługuje się protokołem UDP. Jaki jest w takim razie źródłowy adres IP wysłanego pakietu, skoro Twój komputer jeszcze nie ma IP?

Zadanie 2. Obejrzyj tablicę ARP poleceniem

```
#> arp -n
```

Usuń z niej wszystkie wpisy poleceniami

```
#> arp -d adres_ip_do_usunięcia
```

Zaobserwuj, że w momencie, kiedy wywołujesz polecenie `ping adres_IP`, Twój komputer najpierw wysyła zapytanie ARP, otrzymuje na nie odpowiedź, a następnie wysyła komunikaty *ICMP echo* i otrzymuje na nie odpowiedzi. Sprawdź, że po otrzymaniu komunikatu ARP odpowiedni wpis pojawia się w tablicy ARP.

Obejrzyj dokładnie powyższe komunikaty, zwracając uwagę na to, jakie informacje zapisane są w poszczególnych warstwach oraz na szesnastkową postać ramki. W szczególności sprawdź odpowiedzi na następujące pytania:

- Co jest danymi w przypadku zapytań ARP?
- Jaki jest ethernetowy adres odbiorcy, jeśli wykonasz `ping -b 172.16.255.255`?
- Czy zapytania ARP są wysyłane do konkretnej osoby? A odpowiedzi na nie?

Na końcu zkonfiguruj interfejs `eth0` poleceniami

```
#> dhclient -r eth0
#> ifconfig eth0 0.0.0.0 down
```

Pierwsze polecenie zwalnia dzierżawiony adres IP (potwierdź to w Wiresharku), co jest miłym, choć nie absolutnie koniecznym gestem w stosunku do serwera DHCP i zatrzymuje demona `dhclient`.

Zadanie 3. W tym zadaniu utworzymy konfigurację umożliwiającą testowanie zadania programistycznego o programie `router` na jednym komputerze. Stworzymy sieć złożoną z trzech komputerów, które są połączone ze sobą trzema sieciami w trójkąt.

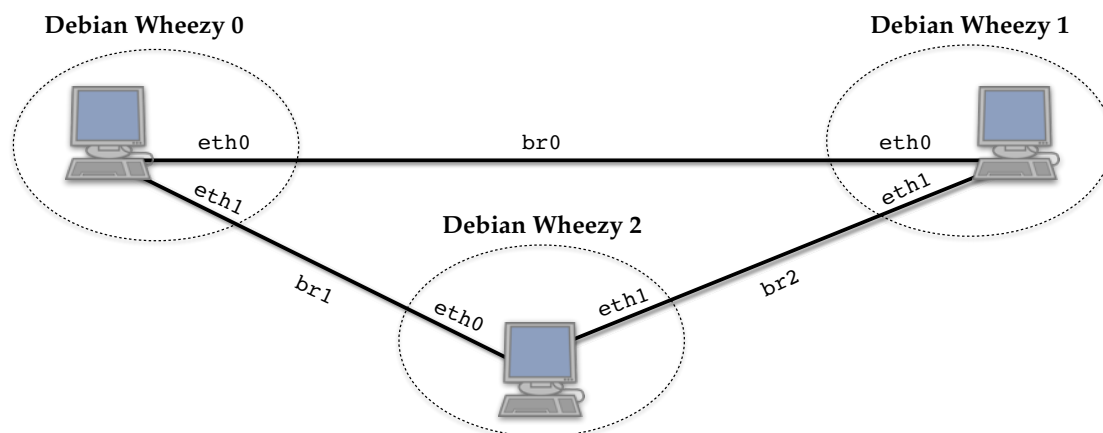
Stwórz wirtualne interfejsy sieciowe, odpowiadające tym trzem sieciom:

```
#> brctl addbr br0
#> ifconfig br0 up
#> brctl addbr br1
#> ifconfig br1 up
#> brctl addbr br2
#> ifconfig br2 up
```

Wyświetl włączone interfejsy sieciowe poleceniem `ifconfig`. Interfejsy `br*` nie mają i nie będą mieć przypisanych adresów IP.

Włącz teraz program `virtualbox` i wybierz w nim maszynę wirtualną *Debian Wheezy 0*. Po prawej stronie kliknij napis *Network* i skonfiguruj jakie interfejsy sieciowe w fizycznej maszynie będą widoczne w maszynie wirtualnej. W tym celu w karcie *Adapter 1* wybierz *Attached to: bridged adapter* a następnie z listy *Name* wybierz `br0`. W zakładce *Adapter 2* wybierz *Attached to: bridged adapter*, zaś na liście *Name* wybierz `br1`. Po uruchomieniu maszyny wirtualnej będzie ona miała dwa interfejsy sieciowe: `eth0` podłączony do interfejsu `br0` maszyny fizycznej i `eth1` podłączony do interfejsu `br1` maszyny fizycznej. Jeśli w maszynie wirtualnej włączony jest `network-manager` należy go wyłączyć poleceniem `/etc/init.d/network-manager stop`.

Powtórz powyższą operację dla drugiej maszyny wirtualnej łącząc jej *Adapter 1* z interfejsem `br0` i *Adapter 2* z interfejsem `br2`, a następnie dla trzeciej maszyny wirtualnej łącząc jej *Adapter 1* z interfejsem `br1` i *Adapter 2* z interfejsem `br2`. Stworzoną konfigurację opisuje poniższy rysunek.



Uwaga: jeśli na maszynie fizycznej brakuje pamięci, warto w maszynie wirtualnej wyłączyć system graficzny. W tym celu przejdź na maszynie wirtualnej do konsoli naciskając

kombinację `RightCtrl + F1` i po zalogowaniu jako użytkownik `root` (hasło znajdziesz w opisie maszyny wirtualnej) wydaj polecenie `/etc/init.d/gdm3 stop`.

Ustal i zapisz na kartce (rozłączne adresowo) adresy IP tych trzech sieci oraz adresy IP poszczególnych kart sieciowych w maszynach wirtualnych. Skonfiguruj je w odpowiednich maszynach wirtualnych poleceniami `ifconfig`, a następnie sprawdź osiągalność połączonych interfejsów i nieosiągalność niepołączonych poleceniem `ping`.

Na końcu wyłącz wszystkie maszyny wirtualne.

Zadanie 4. W tym i kolejnym zadaniu zajmiemy się siecią bezprzewodową. Aktywuj interfejs `wlan0` poleceniem

```
#> ifconfig wlan0 up
```

Poleceniem

```
#> iwlist wlan0 scan
```

wyświetl dostępne punkty dostępowe. Widoczny powinien być punkt należący do instytutowej sieci `eduroam` oraz sieć w pracowni o identyfikatorze `lab109`. Na podstawie wyświetlanych informacji spróbuj odczytać, jakie jest szyfrowanie poszczególnych sieci. Jakie prędkości transmisji obsługują te sieci? Na jakich kanałach nadają te punkty?

Najpierw wykorzystamy program `iwconfig` do konfiguracji warstwy łącza danych łącząc się z punktem dostępowym w pracowni. Zauważ, że w przypadku Ethernetu nie musieliśmy konfigurować warstwy drugiej. Wydadź polecenie

```
#> iwconfig wlan0 essid lab109
```

Poleceniem `iwconfig` sprawdź, że co prawda identyfikator sieci został ustalony, ale nadal nie jesteśmy związani ze stacją bazową (`not associated`). Wynika to z tego, że sieć ta jest zabezpieczona kluczem WPA-PSK, którego jeszcze nie podaliśmy.

Szyfrowanie i uwierzytelnianie WPA obsługuje w Linuksie specjalny demon `wpa_supplicant`. W dowolnym katalogu utwórz plik konfiguracyjny `wpa.conf` z następującą zawartością:

```
ctrl_interface=DIR=/var/run/wpa_supplicant
network={
    ssid="lab109"
    scan_ssid=1
    key_mgmt=WPA-PSK
    psk="bardzotajne"
}
```

Następnie uruchom demona w tle poleceniem

```
#> wpa_supplicant -c wpa.conf -i wlan0 -B
```

Od tej pory warstwa druga jest już skonfigurowana, co można sprawdzić poleceniem `iwconfig`. (Jeśli komputer nadal nie jest związany ze stacją bazową, to nie ma sensu próbować konfigurować warstwy sieci!)

Do konfiguracji warstwy sieci wystarczy wykonać polecenie

```
#> dhclient wlan0
```

Po konfiguracji warstwy drugiej i trzeciej, obejrzyj tablicę routingu i działanie sieci poleceniami `route -n` i `ping`. W parach sprawdźcie również jaką jest przepustowość takiego bezprzewodowego łącza. W tym celu pierwsza osoba powinna uruchomić polecenie

```
$> iperf -s
```

zaś druga polecenie

```
$> iperf -c IP_pierwszego_komputera
```

Jak bardzo spada przepustowość, jeśli uruchamiacie to polecenie wszyscy naraz?

Zakończ demona `wpa_supplicant` wpisując

```
#> wpa_cli terminate
```

a następnie zdekonguruj interfejs `wlan0` poleceniami

```
#> dhclient -r wlan0  
#> iwconfig wlan0 essid off  
#> ifconfig wlan0 0.0.0.0 down
```

Zadanie 5. W tym zadaniu stworzymy sieci *ad-hoc* (tj. bez punktu dostępowego). Włączcie interfejs `wlan0` poleceniem

```
#> ifconfig wlan0 up
```

Połączcie się w kilka grup. (Algorytmy stosowane przez sieci *ad-hoc* w Linuksie nie zawsze działają; grupy dwuosobowe powinny działać bez problemu, z większymi może być problem). Niech każda grupa wybierze sobie nazwę sieci i kanał. Najlepiej wybrać wolny kanał; aktualną listę zajętych kanałów można wyświetlić poleceniem

```
#> iwlist wlan0 scan
```

Wyłącz interfejs poleceniem

```
#> ifconfig wlan0 down
```

a następnie przełącz go w odpowiedni tryb poleceniem

```
#> iwconfig wlan0 mode ad-hoc
```

Ustal nazwę sieci i kanał poleceniami

```
#> iwconfig wlan0 essid nazwa  
#> iwconfig wlan0 freq numer
```

Uaktywnij interfejs poleceniem

```
#> ifconfig wlan0 up
```

Następnie — wewnątrz każdej grupy — przypisz kartom adresy IP z jednej sieci IP i sprawdź czy komputery widzą się nawzajem. Ewentualne problemy można próbować diagnozować poleceniem `iwconfig wlan0`: każda ze stacji w grupie powinna mieć taką samą nazwę ESSID, taką samą częstotliwość i taki sam adres komórki (*cell*). Sprawdź połączenie poleceniem `ping` i przepustowość programem `iperf`.

Lista i materiały znajdują się pod adresem <http://www.ii.uni.wroc.pl/~mbi/dyd/>

Marcin Bieńkowski