

Laboratorium sieci komputerowych

Konfiguracja interfejsów i łącz sieciowych

Artur Skonecki
Łukasz Załęski
gr. 3

18 maja 2012

Streszczenie

Celem zajęć było zapoznanie się z konfiguracją interfejsów i łącz sieciowych.

1 Ethernet

```
-----
|      | .16.16                               .7.3 |      |
| k7 |-----| volt |
|____| fxp0          10.146.0.0/16          lagg0 |____|
```

1.1 Konfiguracja

W celu uniknięcia konfliktów, za pomocą komendy ping sprawdziliśmy czy adres IP nie jest zajęty.

```
k7% ping -c1 10.146.16.16
1 packets transmitted, 0 packets received, 100.0% packet loss
```

Następnie załadowaliśmy sterownik do drugiej karty ethernet

```
kldload ip_fxp
```

Oraz ustawiliśmy adres IP i maskę na interfejsie na fxp0.

```
sudo ifconfig fxp0 inet 10.146.16.16/16 up
```

```
k7% ifconfig fxp0
```

```
fxp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      options=4219b<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, VLAN_HWCSUM, TSO4, WOL_MAGIC, VLAN_HW
      ether 00:0e:0c:00:fc:05
      inet 10.146.16.16 netmask 255.255.0.0 broadcast 10.146.255.255
      media: Ethernet autoselect (100baseTX <full-duplex>)
      status: active
```

W celu przetestowania połączenia wysłaliśmy pakiet ICMP z volta.

```
$ ping -c1 10.146.16.16
PING 10.146.16.16 (10.146.16.16): 56 data bytes
64 bytes from 10.146.16.16: icmp_seq=0 ttl=64 time=0.507 ms
```

Ponadto, nasłuchiwalismy pakiety ICMP za pomocą komendy tcpdump.

```
k7% sudo tcpdump -i fxp0 icmp
20:41:09.371839 IP 10.146.7.3 > 10.146.16.16: ICMP echo request, id 38667, seq 0, length 64
20:41:09.371872 IP 10.146.16.16 > 10.146.7.3: ICMP echo reply, id 38667, seq 0, length 64
```

1.2 Uzyskanie adresu za pomocą DHCP

Następnie ponownie skonfigurowaliśmy interfejs fxp0 za pomocą klienta DHCP.

```
k7% sudo dhclient fxp0
DHCPDISCOVER on fxp0 to 255.255.255.255 port 67 interval 4
DHCPOFFER from 10.146.7.3
unknown dhcp option value 0xaf
DHCPREQUEST on fxp0 to 255.255.255.255 port 67
DHCPACK from 10.146.7.3
unknown dhcp option value 0xaf
bound to 10.146.226.102 -- renewal in 1800 seconds.
```

W celu przetestowania połączenia wysłaliśmy pakiet ICMP z volta do maszyny k7.

```
$ ping -c1 10.146.226.102
PING 10.146.226.102 (10.146.226.102): 56 data bytes
64 bytes from 10.146.226.102: icmp_seq=0 ttl=64 time=0.287 ms
```

Na maszynie k7 tcpdump przechwycił pakiet.

```
k7% sudo tcpdump -i fxp0 icmp
20:53:05.193131 IP 10.146.7.3 > 10.146.226.102: ICMP echo request, id 12568, seq 0, length 64
20:53:05.193159 IP 10.146.226.102 > 10.146.7.3: ICMP echo reply, id 12568, seq 0, length 64
```

2 WiFi

2.1 Połączenie z ZETiIS

Załadowanie sterowników i utworzenie interfejsu wlan0.

```
sudo kldload uhci ehci
sudo kldload if_rum
sudo ifconfig wlan create wlandev rum0
```

```
k7% sudo ifconfig wlan0 up scan
SSID/MESH ID      BSSID              CHAN  RATE   S:N      INT CAPS
ZETiIS            00:1e:52:79:ca:59   6     54M   -75:-95  100 EPS  RSN HTCAP WME
```

Wybranie AP ZETiIS.

```
sudo ifconfig wlan0 ssid ZETiIS
```

Po edycji pliku /etc/wpa_supplicant.conf (ustawienie loginu i hasła) uruchomiliśmy wpa_supplicant.

```
sudo wpa_supplicant -iwlan0 -c/etc/wpa_supplicant.conf
```

Połączenie zostało nawiązane. Stacja k7 odbierała pakiety ARP rozgłaszane jako broadcast.

```
k7% sudo tcpdump -i wlan0
tcpdump: WARNING: wlan0: no IPv4 address assigned
21:59:49.879058 EAP packet (0) v2, len 5
21:59:50.020294 EAP packet (0) v2, len 4
21:59:50.022169 EAPOL key (3) v2, len 117
21:59:50.027094 EAPOL key (3) v1, len 95
21:59:50.208516 IP s04.iem.pw.edu.pl.netbios-dgm > ZET-NET.iem.pw.edu.pl.netbios-dgm: NBT UDP
21:59:50.310614 ARP, Request who-has argo tell vol2, length 46
21:59:50.675298 ARP, Request who-has 192.168.2.70 tell 192.168.2.70, length 46
```

2.2 Połączenie peer-to-peer

Utworzenie interfejsu wlan0 w trybie adhoc.

```
sudo ifconfig wlan0 destroy
sudo ifconfig wlan0 create wlandev rum0 wlanmode adhoc
```

Ustawienie SSID, adresu IP i maski.

```
sudo ifconfig wlan0 ssid K7
sudo ifconfig wlan0 inet 10.7/16
sudo ifconfig wlan0 up
```

Wybraliśmy maszynę k5 w celu przetestowania połączenia z K7.

```
k5% sudo ifconfig wlan0 up scan
SSID/MESH ID      BSSID              CHAN RATE   S:N        INT CAPS
K7                2a:22:6c:6e:06:fa   10   54M -78:-96  100 IS
```

Ustawienie połączenia na komputerze k5.

```
sudo ifconfig wlan create wlandev ath0 wlanmode adhoc
sudo ifconfig wlan0 ssid K7
sudo ifconfig wlan0 inet 10.5/16
```

```
k5% ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 00:0f:cb:ff:13:a7
    inet 10.0.0.5 netmask 255.255.0.0 broadcast 10.0.255.255
    media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
    status: running
    ssid K7 channel 10 (2457 MHz 11g) bssid 2a:22:6c:6e:06:fa
    country US ecm authmode OPEN privacy OFF txpower 22.5 scanvalid 60
    protmode CTS wme burst
```

Test połączenia.

```
k5% ping -c1 10.0.0.7
PING 10.0.0.7 (10.0.0.7): 56 data bytes
64 bytes from 10.0.0.7: icmp_seq=0 ttl=64 time=1.728 ms
```

3 Bluetooth

3.1 Wczytanie sterowników

Załadowanie sterowników usb, bluetooth:

```
kldload uhci ehci
kldload ng_ubt
```

3.2 Uruchomienie systemu bluetooth

Uruchomienie systemu bluetooth dla urządzenia ubt0(modem bluetooth)

```
#/etc/rc.d/bluetooth [start|stop|restart|rcvar] ubt0
/etc/rc.d/bluetooth start ubt0
```

3.3 Uzyskanie dlisty dostępnych urządzeń

Pobranie listy dostępnych urządzeń bluetooth do podłączenia:

```
k6# hccontrol -n ubt0hci inquiry
```

Polecenie zwróciło listę widocznych urządzeń, wraz z ich adresami lub aliasami. Aliasy znajdują się w pliku /etc/bluetooth/hosts Uzyskanie nazwy urządzenia po jego adresie:

```
k6# hccontrol -n ubt0hci remote_name_request 00:08:1b:00:d3:3a
BD_ADDR: K8
Name: k8 (ubt0)
```

3.4 Podłączenie k6 do k8

Włączenie trybu punktu dostępowego (na k8 też ładujemy sterowniki i uruchamiamy system bluetooth):

```
k8# bt-pan -s
```

Po wykonaniu polecenia, konfigurujemy interfejs tap0

```
k8# ifconfig tap0 10.8
k8# ifconfig tap0
tap0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      options=80000<LINKSTATE>
      ether 00:08:1b:00:d3:3a
      inet 10.0.0.8 netmask 255.0.0.0 broadcast 10.255.255.255
```

Następnie podłączamy k6 do punktu dostępowego:

```
k6# bt-pan -c k8
```

Po czym konfigurujemy tap0 na k6

```
k8# ifconfig tap0 10.8
```

Po wykonaniu powyższych operacji sprawdzamy połączenie z k8

```
k6# ping -a k8
44 bytes from K8 seq_no=0 time=38.798 ms result=0
```

4 Łącze szeregowego RS-232

4.1 Konfiguracja

Ponieważ sterowniki były domyślnie wbudowane w kernel FreeBSD, nie było konieczne wykonanie polecenia:

```
kldload uart
```

Znalezienie parametrów łącza obsługiwanych przez obydwa porty.

```
tail /etc/gettytab
```

4.2 Połączenie

Wykonanie na obydwu połączonych maszynach polecenia cu (call unix).

```
k7% sudo cu -l /dev/cuau0 -s 19200
```

```
k8% sudo cu -l /dev/cuau0 -s 19200
```

4.3 Test

Test łącza polegał na wpisaniu ciągu znaków, który był przesyłany na drugi komputer. Zakończenie połączenia za pomocą sekwencji (po dodaniu opcji “EscapeChar none” do ~/.ssh/config):

^M.~

5 Wnioski

Dzięki temu ćwiczeniu zapoznaliśmy się z konfiguracją interfejsów i łącz sieciowych, wykorzystując różne media transmisyjne. Przeprowadziliśmy konfigurację połączeń za pomocą:

- Ethernetu
- WiFi
- Bluetooth
- Łącza szeregowego RS-232
- Firewire

Podczas kładzenia warstwy sieciowej na interfejsy, ważne było wybranie adresu IP z puli adresów prywatnych oraz upewnienie się, że dany adres nie jest już wykorzystywany. W systemie FreeBSD nazwy interfejsów sieciowych są tworzone w zależności od sterownika, który jest używany, co pozwala na łatwiejszą identyfikację fizycznych interfejsów. Ćwiczenie pozwala dostrzec zalety łączy bezprzewodowych, w których wyeliminowanie kabla, jako medium transmisyjnego, pozwala na wygodniejsze i sprawniejsze łączenie urządzeń. W przypadku braku połączenia nie musimy się zastanawiać, czy przypadkiem kabel nie jest uszkodzony.