

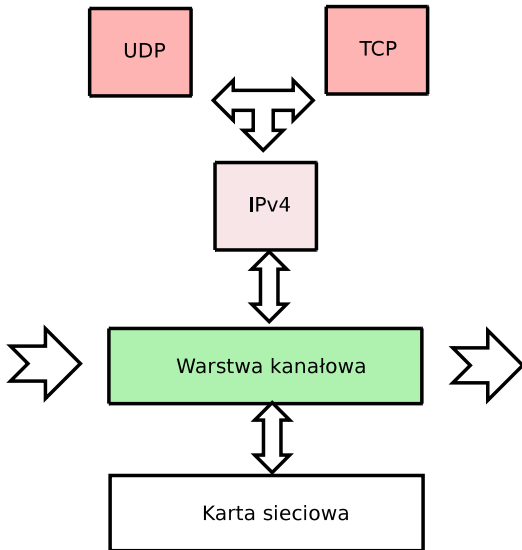
# Warstwa kanałowa

Bartłomiej Świercz

Katedra Mikroelektroniki i Technik Informatycznych

Łódź, 6 maja 2008

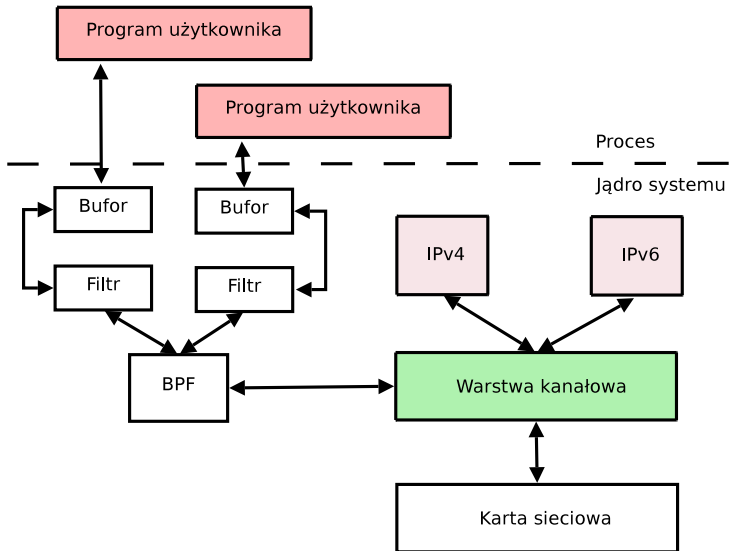
# Warstwa kanałowa



Metody i interfejsy dostępu do warstwy kanałowej:

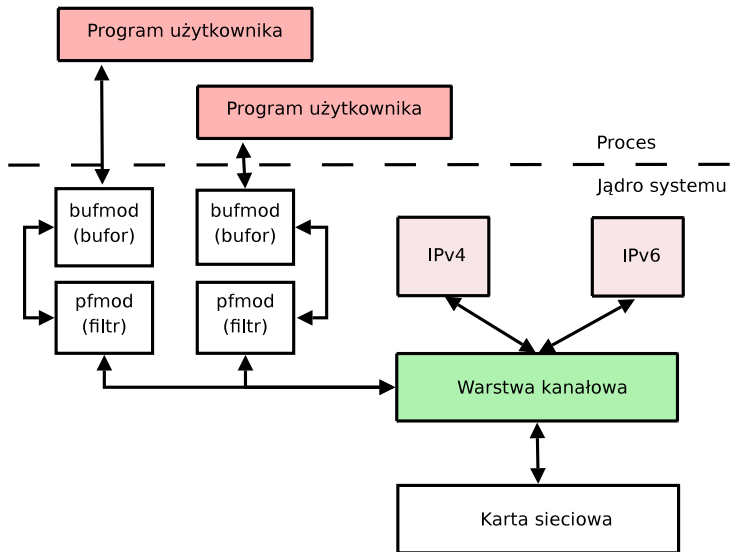
- BPF - BSD Packet Filter
- DLPI - SVR4 Data Link Provider Interface
- Gniazdo SOCKET\_PACKET i PF\_PACKET
- Biblioteka libpcap

# BSD Packet Filter



- filtrowanie wewnątrz jądra,
- przekazuje tylko część każdego pakietu (capture length),
- gromadzi dane w buforze (przekazane przy zapełnieniu lub upłygnięciu czasu czekania na czytanie),
- możliwe jest również wysyłanie pakietów przez BPF (możliwość wysłania innych pakietów niż IP),
- pseudomaszyna BPF działa na acyklicznym grafie skierowanym przepływu sterowania (odwzorowanie na kod dla maszyny rejestrowej).

# SVR4 Data Link Provider Interface



# SVR4 Data Link Provider Interface

- interfejs niezależny od protokołu,
- dostęp poprzez urządzenie np. `/dev/le0`,
- dwa dodatkowe moduły: `pfmod`, `bufmod`,
- filtrowanie na podstawie drzewa wyrażeń boolowskich.

- użytkownik root,
- trzeci argument funkcji socket różny od zera określający rodzaj ramki ethernetowej: ETH\_P\_IP, ETH\_P\_ARP, ETH\_P\_IPV6,
- brak tworzenia bufora w jądrze i filtrowania pakietów,
- dla opcji ETH\_P\_ALL przekazywane będą wszystkie pakiety ze wszystkich urządzeń,
- mechanizm używany przez pierwsze wersje tcpdump,
- metoda dostępu do warstwy kanałowej uznana za przestarzałą (wyparta przez PF\_PACKET).

- użytkownik root lub uprawnienia (capabilities) CAP\_NET\_RAW,
- drugi argument funkcji ma wartość SOCK\_RAW lub SOCK\_DGRAM,
- brak tworzenia bufora w jądrze i filtrowania pakietów
- dla opcji ETH\_P\_ALL przekazywane będą wszystkie pakiety ze wszystkich urządzeń (nagłówek *linux/if\_ether.h*).

# Struktura adresowa sockaddr\_ll

```
#include <linux/if_packet.h>
struct sockaddr_ll
{
    /* PF_PACKET */
    unsigned short    sll_family;
    /* protokół wyższej warstwy */
    unsigned short    sll_protocol;
    /* numer interfejsu używanego do wysłania ramki */
    int                sll_ifindex;
    unsigned short     sll_hatype;
    unsigned char      sll_pkttype;
    /* długość adresu docelowego */
    unsigned char      sll_halen;
    /* adres docelowy */
    unsigned char      sll_addr [8];
};
```

Zapoznamy się z nią wkrótce :)