Gniazda surowe

Bartłomiej Świercz

Katedra Mikroelektroniki i Technik Informatycznych

Łódź, 9 maja 2006

Wstęp

Gniazda surowe posiadają pewne właściwości, których brakuje gniazdom TCP i UDP:

- Gniazda surowe pozwalają na wysyłanie i odbiór pakietów ICMP i IGMP.
- Za pomocą surowych gniazd można przetwarzać datagramy IP, których wartość pola oznaczającego typ protokołu jest nie obsługiwana przez jądro. Pozwala to na realizację obsługi własnego protokołu sieciowego na poziomie użytkownika.
- Przy pomocy gniazd surowych można utworzyć własny nagłówek IPv4, co pozwala na wysyłanie ręcznie spreparowanych pakietów TCP i UDP.

Wstęp

Gniazda surowe posiadają pewne właściwości, których brakuje gniazdom TCP i UDP:

- Gniazda surowe pozwalają na wysyłanie i odbiór pakietów ICMP i IGMP.
- Za pomocą surowych gniazd można przetwarzać datagramy IP, których wartość pola oznaczającego typ protokołu jest nie obsługiwana przez jądro. Pozwala to na realizację obsługi własnego protokołu sieciowego na poziomie użytkownika.
- Przy pomocy gniazd surowych można utworzyć własny nagłówek IPv4, co pozwala na wysyłanie ręcznie spreparowanych pakietów TCP i UDP.

Wstęp

Gniazda surowe posiadają pewne właściwości, których brakuje gniazdom TCP i UDP:

- Gniazda surowe pozwalają na wysyłanie i odbiór pakietów ICMP i IGMP.
- Za pomocą surowych gniazd można przetwarzać datagramy IP, których wartość pola oznaczającego typ protokołu jest nie obsługiwana przez jądro. Pozwala to na realizację obsługi własnego protokołu sieciowego na poziomie użytkownika.
- Przy pomocy gniazd surowych można utworzyć własny nagłówek IPv4, co pozwala na wysyłanie ręcznie spreparowanych pakietów TCP i UDP.

Tworzenie gniazda surowego

1. Należy utworzyć gniazdo ze stałą `SOCK_RAW`:

```
int sockfd;

sockfd = socket (AF_INET, SOCK_RAW, protokół);
```

   Stała oznaczająca protokół może być jedną ze stałych `IPPROTO_xxx`, zdefiniowanych w pliku `<netinet/in.h>`.

2. Można ustawić opcję `IP_HDRINCL`:

```
const int on = 1;

setsockopt (sockfd, IPPROTO_IP, IP_HDRINCL,
            &on, sizeof (on));
```

Tworzenie gniazda surowego

1. Należy utworzyć gniazdo ze stałą `SOCK_RAW`:

```
int sockfd;

sockfd = socket (AF_INET, SOCK_RAW, protokół);
```

   Stała oznaczająca protokół może być jedną ze stałych `IPPROTO_xxx`, zdefiniowanych w pliku `<netinet/in.h>`.

2. Można ustawić opcję `IP_HDRINCL`:

```
const int on = 1;

setsockopt (sockfd, IPPROTO_IP, IP_HDRINCL,
            &on, sizeof (on));
```

Funkcja bind

Dla gniazda surowego można wywołać funkcję `bind()`, któr dowiązuje tylko adres lokalny. Dla gniazd surowych nie istnieje pojęcie portu.

Dla operacji wyjściowych funkcja `bind()` powoduje ustalenie adresu wyjściowego IP, który będzie użyty w datagramach wysyłanych przez surowe gniazdo (nie dotyczy to gniazd z ustawioną opcją `IP_HDRINCL`).

Jeżeli nie wywoła się funkcji `bind()` to jądro jako adres źródłowy ustali adres podstawowego interfejsu sieciowego.

Funkcja connect

Funkcja `connect()` podobnie jak funkcja `bind()` może ale nie musi być użyta. Ponieważ dla gniazd surowych nie istnieje pojęcie portu, to funkcja `connect()` powoduje jedynie przypisanie adresu docelowego do gniazda. Dzięki temu możemy użyć funkcji `write()` i `send()` zamiast funkcji `sendto()`.

Podstawowe reguły wysyłania danych z gniazda surowego

- Operacje wyjścia wykonuje się za pomocą funkcji `sendto()` lub `sendmsg()` i określenia docelowego adresu IP. Jeżeli gniazdo jest połączone to można użyć funkcji `write()`.
- Jeżeli nie ustawiono opcji `IP_HDRINCL` to jądro automatycznie wstawia adres źródłowy do nagłówka IP.
- Jądro dokonuje fragmentacji pakietów surowych, których wielkość jest większa od jednostki MTU dla danego interfejsu sieciowego.
- **W protokole IPv4 proces użytkownika odpowiada za obliczanie i umieszczanie w nagłówku każdej sumy kontrolnej w tym co występuje za nagłówkiem IPv4.**

Podstawowe reguły wysyłania danych z gniazda surowego

- Operacje wyjścia wykonuje się za pomocą funkcji `sendto()` lub `sendmsg()` i określenia docelowego adresu IP. Jeżeli gniazdo jest połączone to można użyć funkcji `write()`.
- Jeżeli nie ustawiono opcji `IP_HDRINCL` to jądro automatycznie wstawia adres źródłowy do nagłówka IP.
- Jądro dokonuje fragmentacji pakietów surowych, których wielkość jest większa od jednostki MTU dla danego interfejsu sieciowego.
- **W protokole IPv4 proces użytkownika odpowiada za obliczanie i umieszczanie w nagłówku każdej sumy kontrolnej w tym co występuje za nagłówkiem IPv4.**

Podstawowe reguły wysyłania danych z gniazda surowego

- Operacje wyjścia wykonuje się za pomocą funkcji `sendto()` lub `sendmsg()` i określenia docelowego adresu IP. Jeżeli gniazdo jest połączone to można użyć funkcji `write()`.
- Jeżeli nie ustawiono opcji `IP_HDRINCL` to jądro automatycznie wstawia adres źródłowy do nagłówka IP.
- Jądro dokonuje fragmentacji pakietów surowych, których wielkość jest większa od jednostki MTU dla danego interfejsu sieciowego.
- **W protokole IPv4 proces użytkownika odpowiada za obliczanie i umieszczanie w nagłówku każdej sumy kontrolnej w tym co występuje za nagłówkiem IPv4.**

Podstawowe reguły wysyłania danych z gniazda surowego

- Operacje wyjścia wykonuje się za pomocą funkcji `sendto()` lub `sendmsg()` i określenia docelowego adresu IP. Jeżeli gniazdo jest połączone to można użyć funkcji `write()`.
- Jeżeli nie ustawiono opcji `IP_HDRINCL` to jądro automatycznie wstawia adres źródłowy do nagłówka IP.
- Jądro dokonuje fragmentacji pakietów surowych, których wielkość jest większa od jednostki MTU dla danego interfejsu sieciowego.
- **W protokole IPv4 proces użytkownika odpowiada za obliczanie i umieszczanie w nagłówku każdej sumy kontrolnej w tym co występuje za nagłówkiem IPv4.**

Pobieranie danych z gniazda surowego

- Odbierane pakiety UDP i TCP **nigdy** nie są przekazywane do gniazda surowego. Jeżeli proces chce poprać pakiety UDP lub TCP to musi to zrobić z warstwy kanałowej.
- **Większość** pakietów ICMP przesyła się do gniazda surowego po zakończeniu przetwarzania komunikatu ICMP przez jądro. Wyjątek to np. ICMP Echo Request.
- **Wszystkie** pakiety IGMP są przekazywane do gniazda surowego po zakończeniu przetwarzania komunikatu przez jądro.
- **Wszystkie** datagramy IP, których pola protokołu mają numer nie znany przez jądro są przekazywane do gniazda surowego. Jądro sprawdza jedynie wersję nagłówka IP, sumę kontrolną IPv4, długość nagłówka i adres docelowy IP.
- Jeżeli datagram przybędzie we fragmentach to zostanie przekazany do gniazda surowego dopiero po otrzymaniu przez jądro wszystkich fragmentów i ich połączeniu.

Pobieranie danych z gniazda surowego

- Odbierane pakiety UDP i TCP **nigdy** nie są przekazywane do gniazda surowego. Jeżeli proces chce poprać pakiety UDP lub TCP to musi to zrobić z warstwy kanałowej.
- **Większość** pakietów ICMP przesyła się do gniazda surowego po zakończeniu przetwarzania komunikatu ICMP przez jądro. Wyjątek to np. ICMP Echo Request.
- **Wszystkie** pakiety IGMP są przekazywane do gniazda surowego po zakończeniu przetwarzania komunikatu przez jądro.
- **Wszystkie** datagramy IP, których pola protokołu mają numer nie znany przez jądro są przekazywane do gniazda surowego. Jądro sprawdza jedynie wersję nagłówka IP, sumę kontrolną IPv4, długość nagłówka i adres docelowy IP.
- Jeżeli datagram przybędzie we fragmentach to zostanie przekazany do gniazda surowego dopiero po otrzymaniu przez jądro wszystkich fragmentów i ich połączeniu.

Pobieranie danych z gniazda surowego

- Odbierane pakiety UDP i TCP **nigdy** nie są przekazywane do gniazda surowego. Jeżeli proces chce poprać pakiety UDP lub TCP to musi to zrobić z warstwy kanałowej.
- **Większość** pakietów ICMP przesyła się do gniazda surowego po zakończeniu przetwarzania komunikatu ICMP przez jądro. Wyjątek to np. ICMP Echo Request.
- **Wszystkie** pakiety IGMP są przekazywane do gniazda surowego po zakończeniu przetwarzania komunikatu przez jądro.
- **Wszystkie** datagramy IP, których pola protokołu mają numer nie znany przez jądro są przekazywane do gniazda surowego. Jądro sprawdza jedynie wersję nagłówka IP, sumę kontrolną IPv4, długość nagłówka i adres docelowy IP.
- Jeżeli datagram przybędzie we fragmentach to zostanie przekazany do gniazda surowego dopiero po otrzymaniu przez jądro wszystkich fragmentów i ich połączeniu.

Pobieranie danych z gniazda surowego

- Odbierane pakiety UDP i TCP **nigdy** nie są przekazywane do gniazda surowego. Jeżeli proces chce poprać pakiety UDP lub TCP to musi to zrobić z warstwy kanałowej.
- **Większość** pakietów ICMP przesyła się do gniazda surowego po zakończeniu przetwarzania komunikatu ICMP przez jądro. Wyjątek to np. ICMP Echo Request.
- **Wszystkie** pakiety IGMP są przekazywane do gniazda surowego po zakończeniu przetwarzania komunikatu przez jądro.
- **Wszystkie** datagramy IP, których pola protokołu mają numer nie znany przez jądro są przekazywane do gniazda surowego. Jądro sprawdza jedynie wersję nagłówka IP, sumę kontrolną IPv4, długość nagłówka i adres docelowy IP.
- Jeżeli datagram przybędzie we fragmentach to zostanie przekazany do gniazda surowego dopiero po otrzymaniu przez jądro wszystkich fragmentów i ich połączeniu.

Pobieranie danych z gniazda surowego

- Odbierane pakiety UDP i TCP **nigdy** nie są przekazywane do gniazda surowego. Jeżeli proces chce poprać pakiety UDP lub TCP to musi to zrobić z warstwy kanałowej.
- **Większość** pakietów ICMP przesyła się do gniazda surowego po zakończeniu przetwarzania komunikatu ICMP przez jądro. Wyjątek to np. ICMP Echo Request.
- **Wszystkie** pakiety IGMP są przekazywane do gniazda surowego po zakończeniu przetwarzania komunikatu przez jądro.
- **Wszystkie** datagramy IP, których pola protokołu mają numer nie znany przez jądro są przekazywane do gniazda surowego. Jądro sprawdza jedynie wersję nagłówka IP, sumę kontrolną IPv4, długość nagłówka i adres docelowy IP.
- Jeżeli datagram przybędzie we fragmentach to zostanie przekazany do gniazda surowego dopiero po otrzymaniu przez jądro wszystkich fragmentów i ich połączeniu.

Pobieranie danych z gniazda surowego - C.D.

Kiedy jądro ma przekazać datagram IP do gniazda surowego to najpierw bada wszystkie otwarte gniazda surowe należące do każdego procesu w celu odnalezienia gniazda spełniającego określone warunki. Kopia pakietu IP przekazana jest do każdego gniazda, które spełnia trzy warunki:

- Sprawdza wartość trzeciego parametru przekazanego do funkcji `socket()` określającego protokół.
- Adres docelowy pakietu IP musi być taki sam jak adres gniazda dowiązany za pomocą funkcji `bind()`.
- Jeżeli adres docelowy był określony dla gniazda za pomocą funkcji `connect()` to adres źródłowy pakietu IP musi być taki sam.

Pytanie: Kiedy przekazane będą do gniazda surowego wszystkie pakiety?

Pobieranie danych z gniazda surowego - C.D.

Kiedy jądro ma przekazać datagram IP do gniazda surowego to najpierw bada wszystkie otwarte gniazda surowe należące do każdego procesu w celu odnalezienia gniazda spełniającego określone warunki. Kopia pakietu IP przekazana jest do każdego gniazda, które spełnia trzy warunki:

- Sprawdza wartość trzeciego parametru przekazanego do funkcji `socket()` określającego protokół.
- Adres docelowy pakietu IP musi być taki sam jak adres gniazda dowiązany za pomocą funkcji `bind()`.
- Jeżeli adres docelowy był określony dla gniazda za pomocą funkcji `connect()` to adres źródłowy pakietu IP musi być taki sam.

Pytanie: Kiedy przekazane będą do gniazda surowego wszystkie pakiety?

Pobieranie danych z gniazda surowego - C.D.

Kiedy jądro ma przekazać datagram IP do gniazda surowego to najpierw bada wszystkie otwarte gniazda surowe należące do każdego procesu w celu odnalezienia gniazda spełniającego określone warunki. Kopia pakietu IP przekazana jest do każdego gniazda, które spełnia trzy warunki:

- Sprawdza wartość trzeciego parametru przekazanego do funkcji `socket()` określającego protokół.
- Adres docelowy pakietu IP musi być taki sam jak adres gniazda dowiązany za pomocą funkcji `bind()`.
- Jeżeli adres docelowy był określony dla gniazda za pomocą funkcji `connect()` to adres źródłowy pakietu IP musi być taki sam.

Pytanie: Kiedy przekazane będą do gniazda surowego wszystkie pakiety?

Pobieranie danych z gniazda surowego - C.D.

Kiedy jądro ma przekazać datagram IP do gniazda surowego to najpierw bada wszystkie otwarte gniazda surowe należące do każdego procesu w celu odnalezienia gniazda spełniającego określone warunki. Kopia pakietu IP przekazana jest do każdego gniazda, które spełnia trzy warunki:

- Sprawdza wartość trzeciego parametru przekazanego do funkcji `socket()` określającego protokół.
- Adres docelowy pakietu IP musi być taki sam jak adres gniazda dowiązany za pomocą funkcji `bind()`.
- Jeżeli adres docelowy był określony dla gniazda za pomocą funkcji `connect()` to adres źródłowy pakietu IP musi być taki sam.

Pytanie: Kiedy przekazane będą do gniazda surowego wszystkie pakiety?

Suma kontrolna IPv4

Sposób liczenia internetowej sumy kontrolnej zdefiniowany jest w dokumencie RFC 1071. Jedna z możliwych implementacji przedstawiona jest po niżej:

```
unsigned short
csum (unsigned short *buf, int nwords)
{
    unsigned long sum;
    for (sum = 0; nwords > 0; nwords--)
        sum += *buf++;
    sum = (sum >> 16) + (sum & 0xffff);
    sum += (sum >> 16);
    return ~sum;
}
```

Długość pola `nwords` jest połową długości nagłówka:

```
iph->ip_sum = controlsum ((unsigned short *) datagram,
                          iph->ip_len >> 1);
```

W ramach wprawki napiszmy odpowiednik programu ping . . .