# Konfigurowanie protokołu PPP

## 1 Wprowadzenie

**Wymagania wstępne: wykonanie ćwiczenia „Statyczny wybór trasy w systemie Linux”.**
Internet jest największą siecią rozległą (WAN). Składa się z wielu sieci lokalnych (LAN), kampusowych (duże sieci LAN, obejmujące od jednego do kilkunastu budynków) i miejskich (MAN). Sieci te najczęściej połączone są ze sobą za pośrednictwem tzw. łączy typu punkt-punkt (ang. point-point link), znajdujących się między ruterami dwóch sieci. Transmisja w tych łączach jest bardzo zróżnicowana – szeregowa albo (rzadko) równoległa, synchroniczna albo asynchroniczna, elektryczna albo światłowodowa. To właśnie łącza typu punkt-punkt czynią Internet siecią rozległą, gdyż łączą odległe od siebie rutery.

Protokołem internetowym dla łączy typu punkt-punkt jest PPP (ang. Point-to-Point Protocol). Zastępuje on wcześniejszy, ciągle jeszcze stosowany protokół SLIP (ang. Serial Line Internet Protocol). PPP został zaproponowany w standardzie RFC 1661; dziś różne jego aspekty opisane są w ponad 50 takich dokumentach.

Na PPP w rzeczywistości składa się cały zestaw protokołów, odpowiedzialnych za tworzenie ramek, uwierzytelnianie oraz negocjację parametrów warstw łącza danych i sieciowej. Poniżej zagadnienia te zostały pokrótce omówione.

### 1.1 Fazy połączenia PPP

Połączenie PPP składa się z czterech faz:

1. Negocjowanie parametrów łącza protokołem LCP (ang. Link Control Protocol). Strony ustalają między innymi prędkość transmisji, maksymalny rozmiar ramki oraz metodę uwierzytelniania. Faza kończy się procesem uwierzytelnienia.
2. Negocjowanie parametrów warstwy sieciowej właściwym protokołem NCP (ang. Network Control Protocol). Wybór protokołu NCP zależy od stosowanego protokołu warstwy sieciowej. Każdemu protokołowi warstwy sieciowej odpowiada jeden protokół NCP, np. IPCP stosowany jest dla IP, IPXCP dla IPX, a DECCP dla DECnet. IPCP umożliwia między innymi dynamiczne ustalenie adresu IP dla jednej ze stron.
3. Wymiana danych.
4. Zamknięcie połączenia.

### 1.2 Uwierzytelnianie: PAP i CHAP

Mechanizm uwierzytelniania PPP zakłada, że klient i serwer współdzielą pewne hasło (ang. secret) lub zbiór par (identyfikator użytkownika, hasło). PPP oferuje do wyboru dwa protokoły uwierzytelniania: PAP (ang. Password Authentication Protocol) oraz CHAP (ang. Challenge-Handshake Authentication Protocol). W pierwszym uwierzytelnienie obejmuje dwie wiadomości i odbywa się tylko jeden raz, z inicjatywy klienta. Przesyła on do serwera parę (identyfikator użytkownika, hasło) w postaci jawnej. Serwer porównuje otrzymane informacje z tymi, które sam przechowuje; jeśli stwierdzi zgodność, uwierzytelnia klienta, a w przeciwnym razie przerywa połączenie. Jak można zauważyć, PAP jest słabym mechanizmem bezpieczeństwa, podatnym na podsłuch, i dlatego nie zaleca się stosowania go na łączach komutowanych.
Lepszą metodę uwierzytelniania stanowi protokół CHAP, w którym przesyłane informacje uwierzytelniające są szyfrowane. Uwierzytelnienie obejmuje tu trzy wiadomości, dokonywane jest okresowo, a jego inicjatorem jest serwer. Najpierw przesyła on do klienta wiadomość (ang. Challange) do zaszyfrowania. Klient po zaszyfrowaniu wiadomości przy pomocy współdzielonego hasła, odsyła ją z powrotem (ang. Response). Serwer porównuje

następnie to, co otrzymał z tym, co sam obliczył i albo potwierdza zgodność, albo przerywa połączenie. W przypadku obydwu protokołów można uwierzytelniać dwustronnie.

### 1.3 Format ramki

Format ramki protokołu PPP przedstawia poniższy rysunek.

| Bajty: | 1 | 1 | 1 | 1 lub 2 | zmienne | 2 lub 4 | 1 |
|---|---|---|---|---|---|---|---|
| | Znacznik | Adres | Kontrola | Protokół | Dane | FCS | Znacznik |

Początek i koniec ramki wyznacza bajt znacznika równy 01111110. Problem pojawiania się tej wartości w polu danych rozwiązano stosując technikę wstawiania bajtów (ang. byte stuffing); bajtem (znakiem) wstawianym przed wartością równą znacznikowi jest 01111101.

Pola Adres i Kontrola również posiadają stałe wartości, kolejno 11111111 i 00000011. Sensowność ich istnienia tłumaczy się tym, że standard przewiduje dla nich inne wartości w przyszłych zastosowaniach protokołu. Póki co jednak obie strony najczęściej negocjują (w fazie 1), by pola te pomijać; pozwala to zaoszczędzić 2 bajty dla każdej ramki.

Pole Protokół określa protokół, którego informacje zawiera pole Dane. Technikę przesyłania komunikatów wielu protokołów w komunikacie innego protokołu nazywa się multipleksacją protokołów (ang. protocol multiplexing). Warto podkreślić, że w ramce PPP kapsułkowane są nie tylko protokoły warstwy trzeciej, ale również wszystkie protokoły związane z samym PPP. Długość pola Protokół, podobnie jak dla pola FCS, jest negocjowana w fazie 1. Pole FCS (ang. Frame Check Sequence) służy do kontroli błędów.

### 1.4 Pliki konfiguracyjne systemu Linux

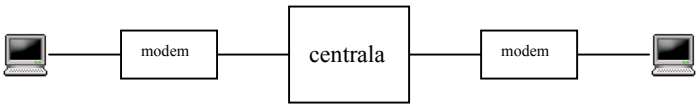
W systemie Linux role klienta i serwera PPP pełni proces systemowy **pppd**. Po uruchomieniu odczytuje on opcje połączenia z pliku **/etc/ppp/options**. Wszystkie opcje opisane są na stronie pppd podręcznika man. W przypadku, gdy stosowane jest uwierzytelnianie, potrzebne informacje należy zapisać w pliku **/etc/ppp/pap-secrets lub /etc/ppp/chap-secrets**.

## 2. Organizacja, wymagany sprzęt i oprogramowanie

- zadania wykonywane są w grupach 2-osobowych;
- sprzęt: 2 komputery PC, 2 modemy, centralka telefoniczna, kabel null-modem dla transmisji szeregowej;
- oprogramowanie: system Linux.

## 3. Zadania

1. Należy połączyć komputery kablem null-modem, a następnie wykonać prostą konfigurację protokołu PPP: bez uwierzytelniania, adres IP każdej ze stron jest znany z góry. Wykorzystać opcje **`netmask`**, **`noauth`** i **`nodetach`** pliku **`/etc/ppp/options`**. Format zapisu adresów IP: <adres lokalny>:<adres zdalny>
2. Zestawić połączenie telefoniczne (bez PPP, tylko fizyczne połączenie) pomiędzy komputerami za pośrednictwem modemów. Należy użyć komend AT modemu (**ATD** do wywołania numeru, **ATA** do przyjęcia wywołania, **ATM** do wyłączenia głośnika modemu; komendy AT omawia załączona dokumentacja). Z modemem łączymy się przy pomocy programu **`minicom`**.



3. Połączyć zadania 1 i 2, zestawiając połączenie PPP za pośrednictwem modemów; role wywołującego i wywoływanego powinien pełnić proces **pppd**. Do ustanowienia połączenia telefonicznego należy wykorzystać program **chat**; W załączeniu znajduje się szablon przykładowej konfiguracji.
4. Włączyć obustronne uwierzytelnianie: najpierw PAP, później CHAP; porównać działanie obydwu protokołów, zapamiętując wcześniej ich komunikaty w pliku logu.

## 4. Pytania sprawdzające

1. Jakie jest zastosowanie protokołu PPP?
2. Jaki jest format ramki protokołu PPP? Dlaczego stosowane są pola Address i Control o stałych wartościach?
3. Do czego służą protokoły LCP i NCP?
4. Wymień i opisz protokoły uwierzytelniania stosowanie w PPP.
5. Na czym polega multipleksacja protokołów?
6. Czy obie strony połączenia PPP muszą posiadać adres IP?

## 5. Literatura

1. Wprowadzenie do protokołu PPP: książki A. S. Tanenbaum „Computer Networks” oraz J. F. Kurose, F. Ross „Computer Networking – A Top-Down Approach Featuring the Internet”.
2. Konfiguracja protokołu PPP w systemie Linux: dokument LINUX HOWTO, strony podręcznika systemu Linux na temat programów pppd i chat.

Plik **/etc/ppp/options** zawiera opcje konfigurujące **pppd** oraz (opcjonalnie) wskazanie do skryptu powłoki, który odpowiada za realizację połączenia, np.:

```
192.168.1.1:
(...)
connect /etc/ppp/connect
```

Przykład 1. Plik **/etc/ppp/options**

Skrypt **/etc/ppp/connect** (musi mieć prawo wykonywania!) może wywoływać program **chat** do realizacji dialogu z modemem, np.:

```
#!/bin/bash
chat –Vef /etc/ppp/dial
#lub chat –Vef /etc/ppp/answer
```

Przykład 2. Skrypt **/etc/ppp/connect**

Program **chat** umożliwia wykonanie skryptów obsługi modemu. Pozwala to na automatyczne zestawienie połączenia modemowego podczas uruchamiania **pppd**. Program **chat** realizuje dialog z modemem w oparciu o plik wskazany przez parametr wywołania.
Struktura pliku jest następująca:

- w każdym wierszu występują dwa pola: pierwsze zawiera ciąg znaków, którego **chat** oczekuje od modemu, a w drugim jest ciąg znaków wysyłany w odpowiedzi do modemu, np.
  **OK ATM0** (na ciąg znaków "OK" otrzymany od modemy wysyłamy komendę wyłączającą głośnik)
  **"" ATZ** (bezwarunkowo zlecamy modemowi wyczyszczenie bieżącej konfiguracji)
- komenda **SAY „tekst"** wyświetla ciąg znaków „tekst" na ekranie lokalnego terminala.

Przykład nr 3 przedstawia skrypt komend AT programu **chat** dla odpowiedzi na żądanie zestawienia połączenia.

```
SAY „Ready to receive a call."
#reset the modem
"" ATZ
OK ATM0
OK ""
RING ATA
CONNECT ""
SAY "\nConnected"
```

Przykład 3. Skrypt **/etc/ppp/answer**