Seriële Laplink HOWTO

Willem J. Atsma

Vertaald door: Ellen Bokhorst

Hoe je tussen twee computers een seriële verbinding tot stand kan brengen. In dit document wordt uitgelegd hoe je een “laplink” verbinding tussen twee computers legt. Zelf ondervond ik wat problemen bij het verkrijgen van de informatie om dit werkend te krijgen, dus dit document is bedoeld het je wat makkelijker te maken.

Table of Contents

1. Introductie

In dit zeer beknopte document wordt uitgelegd hoe een seriële “laplink” verbinding tussen twee Linux machines of een Linux en een Windows machine tot stand te brengen. De seriële verbinding bestaat uit een “null-modem” kabel en een communicatieprotocol: PPP. Ik gebruik een dergelijke setup om bestanden van en naar mijn laptop te versturen; het kan ook prima worden gebruikt om nieuwe software vanaf een cdrom te installeren wanneer de laptop er geen heeft. Ik ondervond meer problemen dan nodig was toen ik mijn seriële laplink instelde. Hopelijk zal dit document er toe bijdragen dat je problemen in een keer zullen worden opgelost. (datgene wat te maken heeft met de seriële kabel).

De setup van de server en client die ik zal beschrijven kan zowel bestemd zijn voor een Linux machine als een ander besturingssysteem, zolang als het ’t PPP-protocol met een null-modem ondersteunt. Waar ’t op uitdraait is dat de MS Windows implementatie speciale aandacht vereist aangezien het iets anders is dan de standaard. Dit document bestaat uit twee hoofddelen. In de eerste wordt de serversetup beschreven. Dit is het hoofdstation met een seriële poort bestemd voor een PPP-verbinding. Als alternatief kan de verbinding wanneer nodig worden gestart vanaf de opdrachtregel. In het tweede deel wordt de configuratie van de client beschreven die een verbinding maakt met de server om bestanden op te halen of naar over te brengen.

Deze documentatie werd geprepareerd met gebruik van pppd versie 2.4.0 met kernelversie 2.4.2.

Laat het me alsjeblieft weten als je specifieke problemen ondervindt en suggesties voor verbeteringen voor dit document hebt; mijn emailadres is aan het einde van dit document te vinden.

2. Op de server

2.1. Configureer pppd

De communicatie zal via de seriële poort verlopen. Op de server zullen we een verbinding maken via het "/dev/ttyS1" (COM2 onder DOS/Windows) device. Dit kan op jouw computer anders zijn en je zult de onderstaande voorbeelden overeenkomstig moeten aanpassen. Ik verplicht tot authenticatie omdat ik de configuratie gebruik om in een manier te voorzien bestanden op een fileserver te benaderen voor een groep mensen. Na het inloggen kunnen zij de gedeelde Samba bestanden gebruiken om bestanden van en naar een laptop te versturen. Als je je geen zorgen maakt om beveiliging dan kun je commentaartekens plaatsen voor de relevante opties.

PPP (Point-to-Point Protocol) communicatie voorziet in TCP/IP over een seriële link. Met andere woorden: wanneer je via een modem op internet gebaseerde browsing wilt, dan zul je dit zeer waarschijnlijk gebruiken. Onder Linux wordt PPP geïmplementeerd door de PPP daemon "pppd". De configuratie ervan vindt plaats via bestanden in de directory "/etc/ppp/". We zullen de volgende bestanden in gebruik gaan nemen:

/etc/ppp/options bevat alle algemene opties voor PPP verbindingen

/etc/ppp/options.ttyS1

    bevat PPP opties voor specifieke verbindingen via "/dev/ttyS1"

/etc/ppp/pap-secrets

    bevat authenticatie informatie

/etc/inittab

    start initiële systeemservices

Het bestand "/etc/ppp/options" moet er ongeveer zo uit komen te zien:

```
#/etc/ppp/options
lock
#auth  dwingt de peer tot authorisatie
#login gebruik voor authenticatie het systeemwachtwoordbestand
#NOTE: mijn pap-secrets staat iedereen toegang toe, dus als
#      dit niet is opgegeven, dan kan iedereen een verbinding!
#      Als dit een machine is waarmee je ook uit kunt bellen,
#      plaats dan commentaartekens voor auth en login en
#      verplaats ze naar /etc/options.ttySn
#
#
auth
login
```

De regels beginnend met een "#" worden aangemerkt als commentaar. De "lock" parameter geeft aan dat een lock bestand zal worden aangemaakt om exclusief toegang tot het seriële device te garanderen, de "auth" parameter geeft aan dat de client zichzelf bekend zal moeten maken en "login" vertelt pppd voor de authenticatie gebruik te maken van de systeemgebruikersnamen en wachtwoorden.

Het bestand "/etc/ppp/pap-secrets" wordt nog steeds gecontroleerd op de gebruikersnaam en wachtwoordinformatie. Een speciale op zichzelf staande regel zorgt dat pppd alleen de systeeminformatie van de gebruiker gebruikt. Het bestand "/etc/ppp/pap-secrets" ziet er ongeveer zo uit:

```
#/etc/ppp/pap-secrets
# Sleutels voor authenticatie met PAP
```

```
# client                server                secret        IP addresses
*                       *                     ""            ppp_laplink_client
```

Elke machine met het IP-adres “ppp_laplink_client” kan nu een verbinding maken met de server, zonder gebruik te maken van een wachtwoord (je kunt zien waarom het handig is de ppp_laplink entries in het bestand /etc/hosts te hebben). De “login” parameter in het bestand “/etc/ppp/options” echter, maakt dat de gebruikersnaam en het wachtwoord geleverd door de client voor authorisatie overeen moeten komen met het bestand “/etc/ppp/pap-secrets” als ook met de systeemgebruikersnaam en het wachtwoord, dus de verbinding zal alleen succesvol zijn nadat in een geldige gebruikersnaam en wachtwoord is voorzien.

Opties die specifiek zijn voor de seriële lijn via welke je een verbinding maakt worden geplaatst in “/etc/ppp/options.ttySn”, waar n het nummer is van het seriële device. device. Mijn server gebruikt “/dev/ttyS1”, dus de opties gaan in... “/etc/ppp/options.ttyS1”.

```
#/etc/ppp/options.ttyS1
asyncmap 0
crtscts
#local geeft aan dat modemlijnen niet worden gebruikt
local
#silent zorgt dat pppd totdat een verbinding is gemaakt vanaf
#de andere kant
silent
#dit zijn regels die voorkomen in het /etc/hosts bestand
#de link werkt niet als dit aan het einde vh bestand staat
#de volgorde doet er dus toe!
ppp_laplink_server:ppp_laplink_client
#auth forceert authorisatie van peer
#login maakt dat voor authenticatie gebruik wordt gemaakt van
# het systeemwachtwoordenbestand
#NOTE: mijn pap-secrets staat iedereen toegang toe, dus als dit
#niet is opgegeven, dan kan iedereen een verbinding maken!
#Als dit een server is die ppp nooit zal gebruiken om uit te
#bellen, dan verplaats je auth en login naar /etc/ppp/options
#auth
#login
#gebruik PAP, geen CHAP voor authenticatie
require-pap
115200
```

Deze opties staan allemaal goed beschreven in de manpage van pppd; een paar van de belangrijkste worden als commentaar in het bestand uitgelegd. De “crtscts” parameter vertelt pppd gebruik te maken van hardware flow control. Dit is aan te bevelen omdat het ’t snelst is. Als alternatief kun je “xonxoff” opgeven om gebruik te maken van op software gebaseerde flow control - je geeft dit op als je nullmodem kabel geen verbinding maakt met de RTS/CTS lijnen (onwaarschijnlijk als je je kabel in de winkel hebt gekocht). “115200” duidt op de snelheid van de gegevenstransmissie - als je problemen ondervindt bij het tot stand brengen van de verbinding, zou je een lagere snelheid kunnen proberen. Geldige snelheidsinstellingen zijn te vinden in de manpage van termios, alhoewel je hardware een beperking zal opleggen voor de baud rate. Het instellen van een ongeldige snelheidsinstelling zal een foutmelding teweeg brengen van de pppd daemon en blijkbaar valt het dan terug op een standaardwaarde.

Voor de opties “auth” en “login” zijn hier commentaartekens geplaatst, omdat ze in het algemene options bestand werden opgegeven. Als je je computer ook gebruikt om bij een ISP in te bellen, dan geef je ze hier op in plaats van in “/etc/ppp/options”, anders vraag je je ISP namelijk om zichzelf te authoriseren wanneer je inbelt wat waarschijnlijk geen succes zal hebben. De reden waarom ze niet in dit bestand worden gespecificeerd is dat wanneer je nu of in de toekomst andere inkomende PPP-verbindingen hebt, je er zeker van wilt zijn dat ze altijd zijn geautoriseerd. Denk eraan dat de hier gepresenteerde “pap-secrets” geen enkele bescherming geeft.

Ten slotte specificeert de “ppp_laplink_server:ppp_laplink_client” regel de lokale en het remote IP-adres nadat de link in de lucht is. Je kunt hier echte IP nummers gebruiken (b.v. 192.168.0.1:192.168.1.1), of regels uit het bestand “/etc/hosts”, zoals ik deed. Het mooie van dat laatste is dat je de namen kunt gebruiken om later naar deze links te refereren. Ik raad je ook aan IP-nummers te gebruiken zoals ik dat deed (192.168.0.1:192.168.1.1). Deze adressen zijn gereserveerd voor lokale netwerken en ze bestaan niet op het internet, dus je voorkomt mogelijke conflicten. Als de link eenmaal is geactiveerd, kan de client verwijzen naar de server met het IP-adres van ppp_laplink_server (192.168.0.1) en de server verwijst naar de client met ppp_laplink_client (192.168.1.1).

Je kan verschillende regels gebruiken in “/etc/ppp/pap-secrets” om alleen gebruikerstoegang te selecteren. Ik gebruik het PAP protocol voor authenticatie; als je dat wilt, kun je CHAP gebruiken - de setup is vrijwel gelijk, waarbij je gebruik maakt van het bestand “chap-secrets”. Voor deze en andere opties kun je de manpages en de documentie die onderaan het document wordt vermeld raadplegen.

2.2. Een op getty lijkende installatie van pppd

Je kunt de PPP-daemon (pppd) starten wanneer je het systeem boot en het de de seriële lijn van je keuze laten monitoren. Een elegante manier om dit te bereiken is het wijzigen van het bestand “/etc/inittab”. Dit bestand bevat informatie voor initialisatie van het systeem. Voeg het volgende toe aan dit bestand:

```
# Start pppd voor de seri&euml;le laplink.
pd:2345:respawn:/usr/sbin/pppd /dev/ttyS1 nodetach
```

Dit betekent het volgende: voor runlevels 2, 3, 4 en 5 “/usr/sbin/pppd /dev/ttyS1 nodetach” starten en als het wordt beëindigd (einde van de verbinding) start dan een nieuwe op. De “nodetach” option maakt dat pppd verbonden blijft met de terminal welke het startte, in plaats van dat het forkt en stopt. Deze optie is nodig omdat het “init” proces anders onmiddellijk een nieuwe verbinding op zou starten. Andere regels in het inittab bestand specificeren getty processen die op seriële terminals (tty’s) draaien; die initialisatie lijkt veel op deze.

Typ voor het activeren van deze nieuwe configuratie:

```
[root@griis /root]# /sbin/init q
```

2.3. Start de server wanneer dit nodig is

Als je slechts zo nu en dan een verbinding met de server wilt maken, geef je er waarschijnlijk de voorkeur aan de verbinding handmatig op te starten. Alle instellingen blijven hetzelfde; je kunt de server eenvoudigweg starten door achter de prompt in te tikken:

```
/usr/sbin/pppd /dev/ttyS1 nodetach
```

De optie “nodetach” is niet echt nodig, maar zorgt dat de verbinding makkelijk kan worden afgebroken door “ctrl-c” in te drukken.

2.4. Diensten aanbieden aan MS Windows clients

Helaas is de MS Windows implementatie niet geheel standaard. Voor het initiëren van de PPP-verbinding vereist het de uitwisseling van de tekststrings “CLIENT” (van de client) en “CLIENTSERVER” (van de server). Voor aanpassing aan een Windows client moet de volgende regel worden toegevoegd aan het bestand “/etc/ppp/options.ttyS1”:

```
connect 'chat -v -f /etc/ppp/scripts/winclient.chat'
```

Maak dan de scripts directory en het chat bestand “/etc/ppp/scripts/winclient.chat” aan:

```
TIMEOUT 3600
CLIENT CLIENTSERVER\c
```

De connect optie maakt het je mogelijk dat je een programma opgeeft die de string uitwisseling verzorgt voor de connectie. Gewoonlijk wordt het programma “chat” hiervoor gebruikt; kijk in de manpage voor details. Het gegeven script zorgt voor wat nodig is voor de Windows connectie. Je hebt het niet nodig om een verbinding te maken met een Linux box.

3. Op de client

Na de server te hebben geconfigureerd, is de client een makkie. In het bestand “/etc/ppp/options” komt een enkele regel voor:

```
#/etc/ppp/options
lock
```

De seriële poort op mijn client is “/dev/ttyS0” (COM1), dus moet ik ook een bestand “/etc/ppp/options.ttyS0” aanmaken:

```
#/etc/ppp/options.ttyS0
115200
crtscts
local
user zaphod
noauth
```

Slechts een paar nieuwe opties in dit bestand. Merk op dat de snelheid op de server en client hetzelfde moet zijn (hier is dat 115200). Met “user zaphod” is de gebruikersnaam op de client gespecificeerd. Deze gebruikersnaam moet corresponderen met een regel in het bestand “/etc/ppp/pap-secrets” van de client en het moet een geldige gebruikersnaam op de server zijn. De optie “noauth” geeft aan dat de peer (server) zichzelf niet bekend hoeft te maken aan de client.

Als laatste: het instellen van de clientnaam (gebruikersnaam) en de sleutel (het wachtwoord) in het bestand “/etc/ppp/pap-secrets”:

```
#/etc/ppp/pap-secrets
# Secrets for authentication using PAP
# client                server                secret                IP addresses
zaphod                  *                     gargleBlaster
```

Dus de gebruiker “zaphod” gebruikt het wachtwoord “gargleBlaster”, wat hem toegang zou moeten geven tot het systeem. Deze bestanden bevatten gevoelige informatie, dus zorg dat je de permissies correct instelt.

3.1. Start de client

Typ op de opdrachtregel:

```
/usr/sbin/pppd /dev/ttyS0 nodetach
```

3.2. Verbinding maken met een MS Windows server

Wat als je een verbinding wilt maken met een Windows server? Ook hier hebben we de optie “connect” nodig en een chatscript. Voeg de volgende regel toe aan het bestand “/etc/ppp/options.ttyS0”:

```
connect chat -v -f /etc/ppp/scripts/winserver.chat
```

Maak ook het chatscript “/etc/ppp/script/winserver.chat” aan:

```
TIMEOUT 10
■ CLIENT\c
```

Dit zorgt ervoor dat de client de “CLIENT” string stuurt voordat geprobeerd wordt de PPP-verbinding te starten.

3.3. Instellen van een MS Windows (95) client

Om met windows een verbinding te maken met de server, gebruik je het programma Directe kabelverbinding, en stel je het in als “guest”. Om met de windows client op volledige snelheid een verbinding te maken, zul je de communicatiesnelheid van de directe kabelverbinding moeten wijzigen. Hiervoor ga je naar Control Panel->Systeem->Device Manager->Modems en selecteert de seriële kabel. Ga verder met de ’modem’ optie en selecteer een maximumsnelheid van 115200 of waarop je linux computer dan ook is ingesteld. Nu is het van belang dat je in dit stadium je computer opnieuw opstart, anders zal het niet werken (tenminste niet met Microsoft Windows 95, versie A), wat betreft andere versies ben ik daar niet zeker van.

3.4. Vreemd genoeg...

Vreemd genoeg creërde ik een Linux server die een verbinding met zowel Linux als Windows computers tot stand moest brengen, dus bevond ik mezelf in de vreemde situatie beide Linux systemen zodanig te moeten instellen dat ze zowel een verbinding maakten met Windows servers als clients!

4. Verbinding maken...

Het moment van de waarheid: verbind de computer met de nullmodem kabel, start als eerste de server (als je de de “/etc/inittab” benadering gebruikt, moet het reeds zijn geactiveerd), en start dan de client. Op de server krijg je iets dergelijks te zien als het volgende:

```
[root@griis /root]# pppd /dev/ttyS1 nodetach
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS1
user zaphod logged in
Deflate (15) compression enabled
local  IP address 192.168.0.1
```

```
remote IP address 192.168.1.1
```

Als je pppd opstartte vanuit het bestand "/etc/inittab" dan zullen deze meldingen verschijnen in "/var/log/messages". Op de client zie je dan iets als:

```
[root@wylg /root]# pppd /dev/ttyS0 nodetach
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS0
Remote message: Success
Deflate (15) compression enabled
local  IP address 192.168.1.1
remote IP address 192.168.0.1
```

Op beide computers kun je de andere computer pingen om erachter te komen of de verbinding actief is:

```
[zaphod@wylg zaphod]$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) from 192.168.1.1 : 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=0 ttl=255 time=31.7 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=20.3 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=19.2 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=20.3 ms

--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 19.2/22.8/31.7 ms
```

Je kunt nu de server via alle TCP/IP services benaderen die het beschikbaar heeft, zoals FTP, HTTP, TELNET en SAMBA.

5. Internet via de seriële poort

Veronderstel dat je een laptop hebt en zo nu en dan een verbinding wilt maken met Internet. Stel ook dat je een machine hebt met een ethernet verbinding en een seriële poort. Je kunt de laptop als een client instellen, de andere machine als een server en IP masquerading gebruiken om via de null-modem kabel een verbinding te maken met Internet. Kijk de IP-Masquerading-Howto er op na voor hoe je dit kunt doen.

6. Andere nuttige documenten

- De pppd en chat manual pages
- PPP-HOWTO
- How to hook up PPP (http://axion.physics.ubc.ca/ppp-linux.html)
- Modem-HOWTO
- Serial-HOWTO
- IP-Masquerade-HOWTO

7. License

8. Krediet

Ik heb een aardige hoeveelheid behulpzame feedback ontvangen, in het bijzonder van J.M. en B.S., en een deel daarvan is opgenomen in deze nieuwe versie, bedankt! author: W.J. Atsma email: watsma@mech.ubc.ca revision: 29jul2001