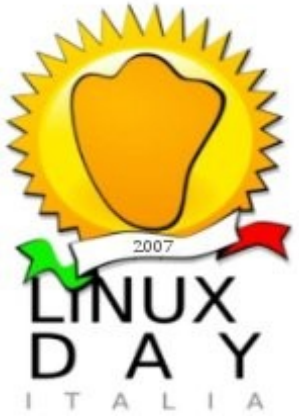
Mesh Node Building with La Fonera

LinuxDay Roma 2007



NINUX.ORG

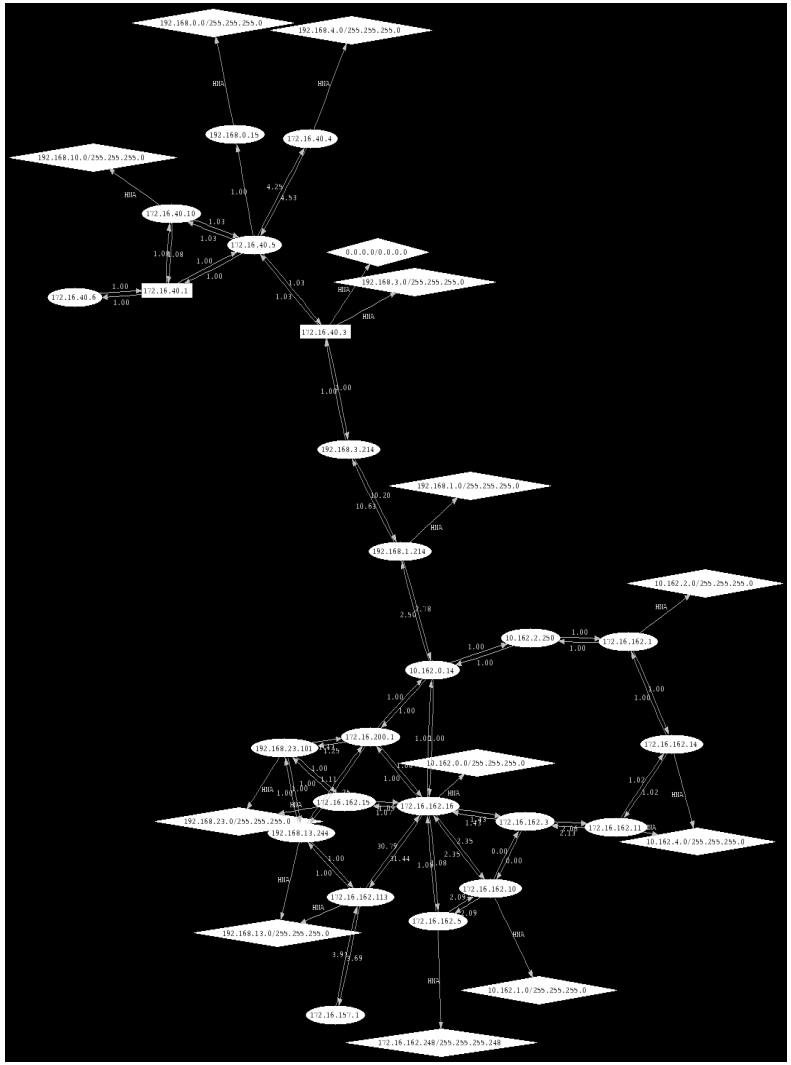
Ninux.org

wireless network community Roma

27/10/2007

Ninux.org – Chi siamo

Ninux.org è una community che ha lo scopo di realizzare a Roma una rete wireless libera, senza scopi di lucro, e nel rispetto della logica open source.



E' necessario:

Gestione Indirizzi IP univoci
(chi sono gli altri?)

Routing -> nel nostro caso OLSR
(dove sono gli altri?)

Metrica Radio Aware -> ETX
(uso dei percorsi migliori)

Tecnologia Radio ->IEEE 802.11 Ad-Hoc
(può essere anche altro!)

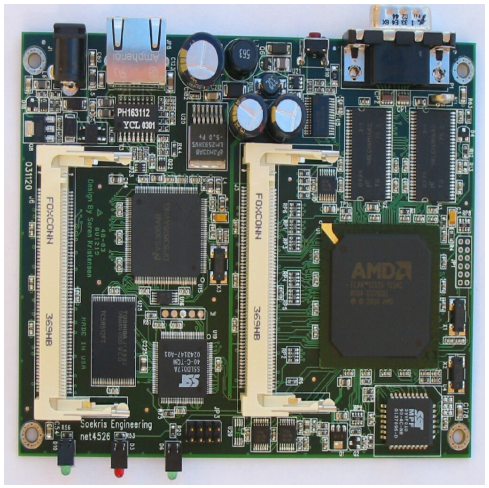
More info**: http://wiki.ninux.org/ChiSiamo**

Creare un Nodo

In questo talk ci occuperemo di spiegare la messa in funzione di un nodo tramite queste soluzioni:

- **Alimentazione:** POE(Power Over Ethernet)
- **Hardware**:La Fonera
- **Firmware:** OpenWRT Kamikaze
- **Routing:** OLSR

Ogni router wireless che supporti la configurazione ad-hoc e olsr è un potenziale nodo della rete ninux

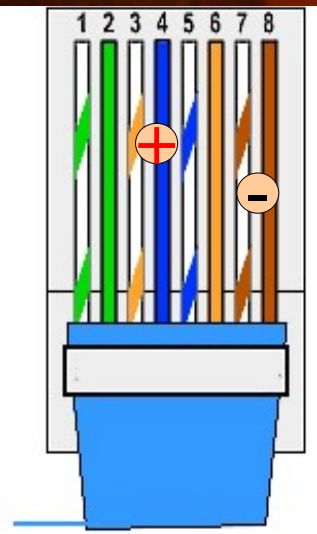


More info:**http://wiki.ninux.org/NuovoNodo**

Alimentazione - PoE

PoE --> Power over Ethernet

Porta l'alimentazione sul cavo ethernet



More info:**http://wiki.ninux.org/PowerOverEthernet**

Hardware - La Fonera

La Fonera è un router wireless **802.11b/g** basato su una cpu **Atheros.** Il firmware con cui viene inviata è una versione modificata da **FON** di **OpenWrt** .

- CPU: Atheros AR531X_COBRA - MIPS 4KEc V6.4 - 183.50 mhz
- RAM: Hynix hy57v281620etp-h - 16 MB
- Flash: ST(84) H - 25P64V6P - MYS 636 - 8 MB
- Ethernet: (1x) Altima AC101 (10/100 Mbit/s) [Auto-MDI(X)]
- Wireless: IEEE 802.11b / 802.11g (up to 54 Mbps)
- Antenna Connector: RP-SMA Connector (Reverse SMA)
- Antenna Omni-Directional detachable antenna (2dBi)
- Dimensions: 93.5 mm x 25.5 mm x 70 mm (excluding antenna)
- Powersupply Input: 100-240V ~ 50-60 Hz 0.3A. Output: 5V DC, 2.0A Output
- Power Consumption: 4 Watt
- Reset Switch

More info: **http://wiki.ninux.org/LaFonera**

Firmware - OpenWRT

OpenWrt è un firmware basato su GNU/Linux specifico per dispositivi embedded basati su chipset Broadcom. Il supporto originariamente era limitato per i Linksys WRT54G ma ora il supporto è stato allargato anche per altre architetture(LaFonera atheros)

Protocollo di routing - OLSR

OLSR --> Optimized Link State Routing Protocol

OLSR è un protocollo di routing adatto a reti Mesh.

L'implementazione è opensource, disponibile per
•Linux
•BSD
•MAC OS X
•Windows

Ogni nodo è a conoscenza dell'esatta topologia di tutta la rete.
Ha una riga nella sua tabella di routing per ogni altro nodo della rete. Non esistono subnets!
La metrica è radio aware, si chiama ETX (expected trasmissions count)

Vantaggi:
- Autoconfigurabilità
- Migliore scelta dei percorsi su canale radio

Svantaggi:
- Sicurezza

Creazione di un nuovo nodo

1 Step: **Procuriamoci una fonera..**
costo attuale: EUR34.44

Ci arriva a casa così:

Creazione di un nuovo nodo

2 Step: Sostituzione fw fon con KAMIKAZE

Ci sono due metodi per riflashare La Fonera

- **Hack dei DNS**
- **HTTP Code Injection**

Creazione di un nuovo nodo

2 Step: Sostituzione fw fon con KAMIKAZE

1 fase: accesso ssh alla fonera

Via dns:
Hack che funziona anche coi firmware FON più nuovi sfrutta il fatto che la configurazione del demone per l'hotspot ChiliSpot viene scaricata in chiaro e senza alcun controllo dal server RADIUS di FON. E' dunque possibile ”mettersi in mezzo” ed inviare al router una configurazione particolare, che al riavvio esegua il demone SSH.
Per farlo è semplicissimo, basta impostare come DNS questo indirizzo: 88.198.165.155 (corrispondente all'host kolofonium.datenbruch.de)

Via code injection:

Per fare questo è necessario inviare dei comandi in POST all'interfaccia di amministrazione, cosa che si può facilmente fare tramite il proprio browser ed una pagina HTML apposita.Funziona solo con Firmware 0.7.1 r1 e r2.

Creazione di un nuovo nodo

2 Step: Sostituzione fw fon con KAMIKAZE

2 fase: accesso a Redboot

Cosa è **Redboot**?

Redboot è un completo sistema di bootstrap per piattaforme embedded (sviluppato da Red-hat)

Consente di uploadare via tftp file e gestire connessioni via lan nonchè naturalmente riflashare la fonera che è quello che servirà a noi...

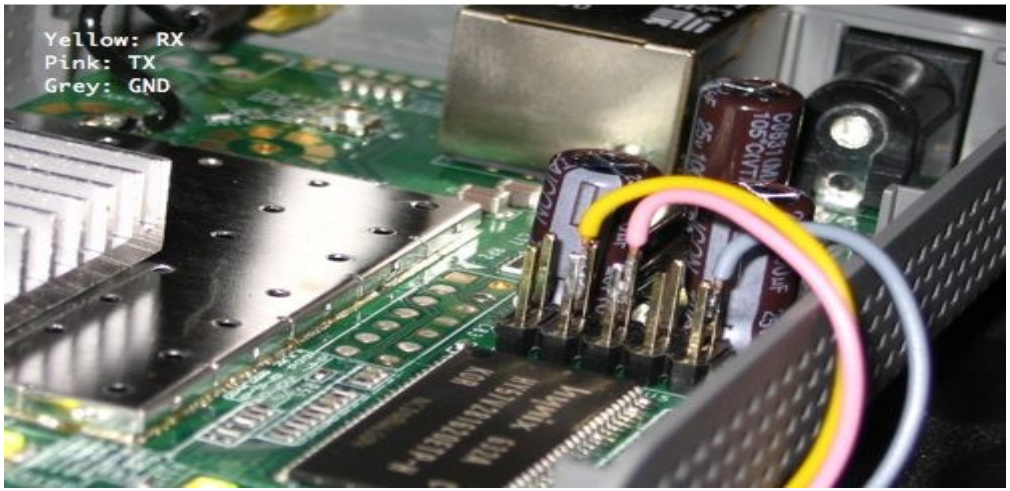
Creazione di un nuovo nodo

2 Step: Sostituzione fw fon con KAMIKAZE

2 fase: accesso a Redboot

Via seriale:

Sulla fonera è disponibile un accesso seriale attraverso dei pin. Realizzando un circuito adattatore TTL-RS232 possiamo interfacciarci direttamente con redboot che è il boot della fonera e poter riflashare ques'ultima



Creazione di un nuovo nodo

2 Step: Sostituzione fw fon con KAMIKAZE

2 fase: accesso a Redboot

Via fw:

Oltre che via seriale è possibile accedere a Redboot anche senza un intervento hw questo grazie al caricamento via ssh dell'immagine realizzata da Camicia che consente di sbloccare l'accesso a Redboot via telnet.

2 Step: Sostituzione fw fon con KAMIKAZE

3 fase: Utilizzo di Redboot

Come ci sei presenta redboot?

```
+PHY ID is 00xx:xxxx
Ethernet eth0: MAC address 00:18:xx:xx:xx:xx
IP: 192.168.1.60/255.255.255.0, Gateway: 0.0.0.0
Default server: 192.168.1.3

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version v1.3.0 - built 16:57:58, Aug  7 2006

Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.

Board: ap51
RAM: 0x80000000-0x81000000, [0x80040450-0x80fe1000] available
FLASH: 0xa8000000 - 0xa87f0000, 128 blocks of 0x00010000 bytes each.
== Executing boot script in 6.000 seconds - enter ^C to abort
```

Creazione di un nuovo nodo

2 Step: Sostituzione fw fon con KAMIKAZE

3 fase: Utilizzo di Redboot

Come ci sei presenta redboot?

Premendo ctr-c otteniamo la shell di **redboot** e da lì attraverso un server tftp attivato sul computer a cui è collegata la fonera possiamo passare con questi comandi il kernel e il filesystem alla fonera:

```
ip_address - l ip.della.fon.era/24 -h ip.adress.del.tftpserver
load -r -v -b 0x80040450 rootfs.squashfs
fis create -b 0x80040450 -f 0xA8030000 -l 0x00700000 -e 0x00000000 rootfs
load -r -b %{FREEMEMLO} kernel.lzma
fis create -r 0x80041000 -e 0x80041000 vmlinux.bin.l7
fis load -l vmlinux.bin.l7
reset
```

Creazione di un nuovo nodo

3 Step: Primo Boot

Non c'è una password di default

Primo login telnet senza password, poi ssh

Non c'è un interfaccia web based

E' un progetto separato, si installa come pacchetto e si chiama webif

```
echo src X-Wrt http://downloads.x-wrt.org/xwrt/kamikaze/7.07/atheros-2.6/packages
>> /etc/ipkg.conf

ipkg update
ipkg install webif
ipkg install webif-lang-it
```

3 Step: Installazione del software

2 fase:Configurare Ipkg

Ipkg è un package managere utilizzato per scaricare e installare i pacchetti di OpenWrt dalla rete (in maniera molto simile ad apt-get di debian)

Aggiungere i repo di X-Wrt

```
echo src X-Wrt http://downloads.x-wrt.org/xwrt/kamikaze/7.07/atheros-2.6/packages
>> /etc/ipkg.conf
```

Verificare che il file sia in ordine

```
root@OpenWrt:~# cat /etc/ipkg.conf
src release http://downloads.openwrt.org/kamikaze/7.07/atheros-2.6/packages
src packages http://downloads.openwrt.org/kamikaze/packages/mips
src X-Wrt http://downloads.x-wrt.org/xwrt/kamikaze/7.07/atheros-2.6/packages
dest root /
dest ram /tmp
root@OpenWrt:~#
```

Aggiornare la lista pacchetti

```
ipkg update
```

Creazione di un nuovo nodo

3 Step: Installazione del software

3 fase:Installare il necessario

Webif(Interfaccia Web)
ipkg install webif,webif-lang-it

Olsr
ipkg install olsrd,olsrd-mod-bmf,olsrd-mod-httpinfo

Avahi
ipkg install howl-mdnsresponder,howl-utils

4 Step: Configurare Olsr

Una volta installato olsr bisognera modificare alcuni file per renderlo operativo.

vi /etc/olsrd.conf

- settare UseHysteresis su no
- settare LinkQualityLevel 2
- settare Interface ath0

Aggiungere le seguenti righe al file /etc/config/network

```
config interface     "ninuxif"
        option ifname    "ath0"
        option proto     "static"
        option ipaddr    "172.16.CAP.1"
        option netmask   "255.255.0.0"
        option gateway   ""
        option dns       "160.80.2.5"
```

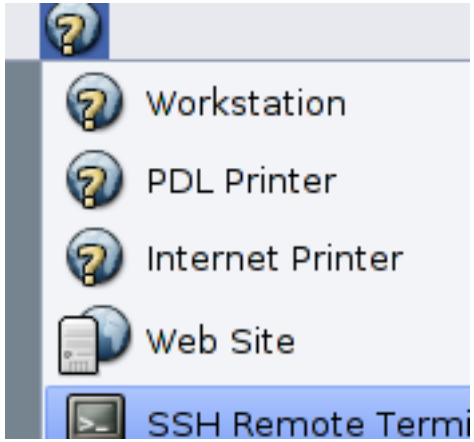
Far partire *olsrd* all'avvio

```
ln -s /etc/init.d/olsrd /etc/rc.d/S95olsrd
```

Annunciare Servizi con Avahi

Avahi è un implementazione opensource di **zeroconf** sistema che facilita l'individuazione e la pubblicazione di servizi all'interno di una rete.

I servizi vengono annunciati dai server utilizzando il **Multicast**, non esiste quindi un server centrale ma **tutto è distribuito**.

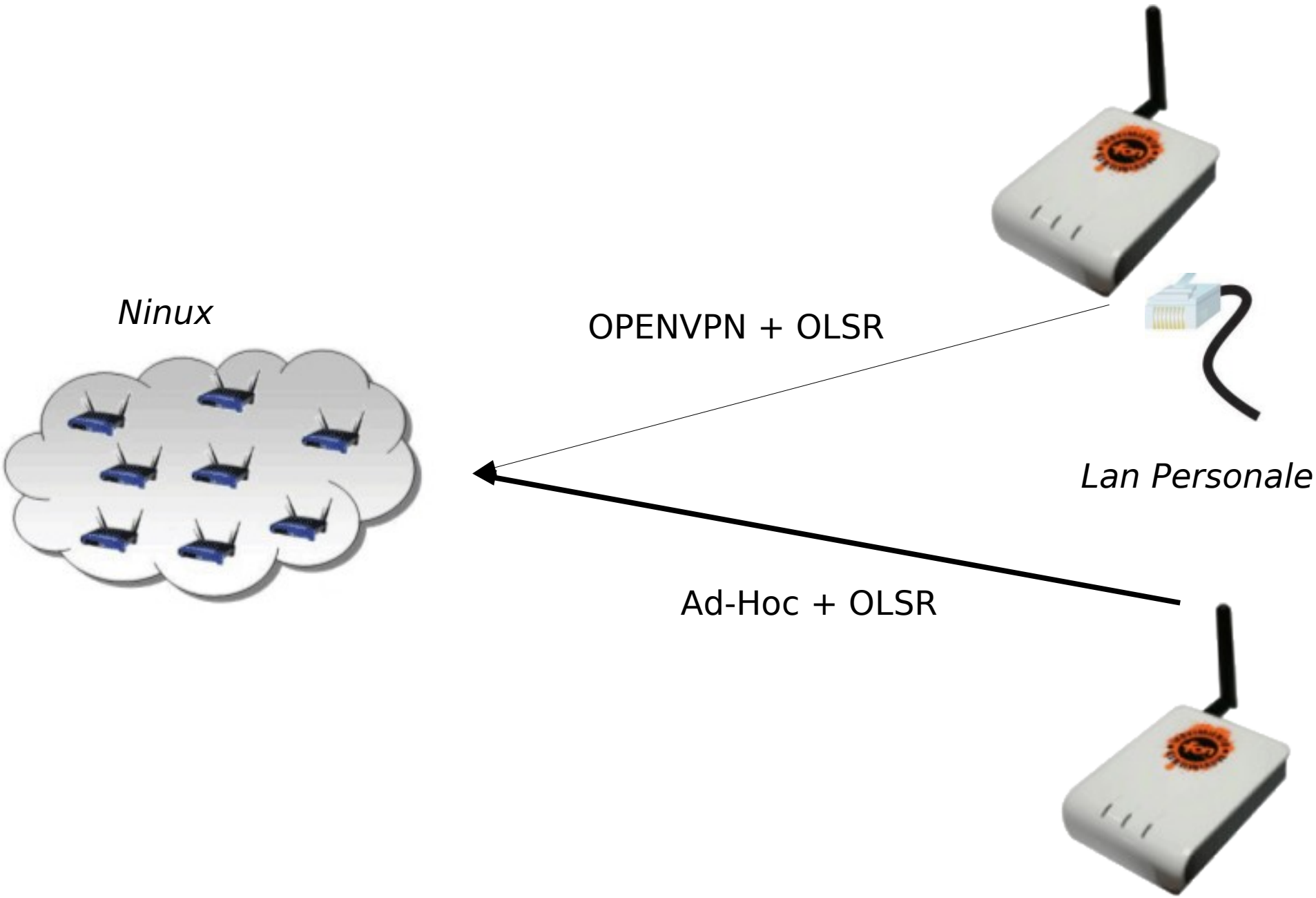


Tra gli utilizzi più comuni di questa tecnologia si ha iTunes(ma anche Rhythmbox,Banshee) e le stampanti di rete ma generalmente si può annunciare qualsiasi tipo di servizio.

Per annunciare servizi con la fonera si deve editare il file /etc/mDNSResponder.conf e avviare il demone mDNSResponder.

```
root@OpenWrt:~# cat /etc/mDNSResponder.conf
#name          #type          #domain  #port  #text
"Fonera"  _workstation._tcp  local.  9   "txtvers=1" "note=La Fonera di eugenio"
"Fonera Webif"   _http._tcp local.  80  "txtvers=1""path=/P" "note=Fon Kamikaze"
root@OpenWrt:~#
```

Join Us



Mesh Node Building with La Fonera

http://wiki.ninux.org

Ninux.org

wireless network community Roma

27/10/2007