

# Hálózatok építése és üzemeltetése

OpenWRT

# Mai téma

---

- ▶ Hálózati eszközök belső felépítése
  - ▶ egy konkrét példán keresztül
- ▶ OpenWRT
- ▶ (Edge, core routerek → következő előadás)

# OpenWRT történet

# OpenWRT koktélok

---

- ▶ OpenWRT: egy GNU/Linux disztribúció
  - ▶ beágyazott rendszerekhez
  - ▶ “főleg” wireless routerekhez
- ▶ Történet
  - ▶ Linksys WRT54G eszköz: nyílt lett a forráskód (GPL)
  - ▶ 2004
    - ▶ ezekre a GPL-es forrásokra
    - ▶ és az uClibc projektből átvett buildroot környezetre alapozva megjelent az első ún. “stable release”
    - ▶ buildroot környezet: különböző beágyazott rendszerekre (melyek különböző architektúrára és CPU-ra épülnek) tudunk operációs rendszert / firmware-t “készíteni”
      - funkciók: keresztfordítás (cross-compilation), a root filesystem és a megfelelő kernel generálása, bootloader image létrehozása

# OpenWRT koktélok

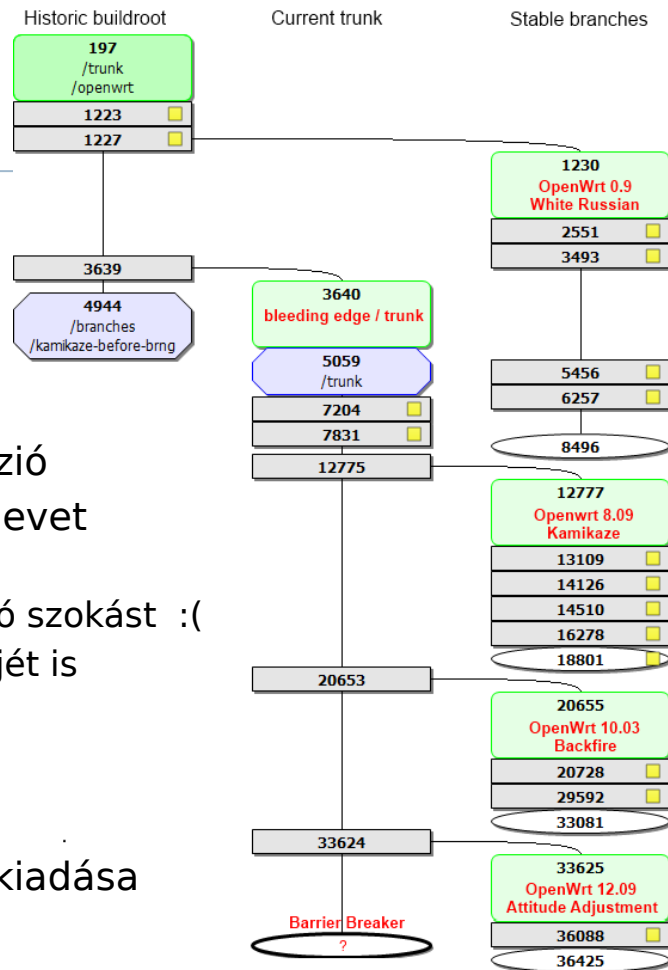
## ▶ Történet

### ▶ 2005

- ▶ sok új fejlesztő csatlakozott a projekthez
- ▶ megjelentek az ún. “experimental” verziók
- ▶ ennek eredménye lett az első önálló OpenWRT verzió
- ▶ ami egy népszerű koktélról a **White Russian** kódnevet kapta
  - a kódnevekkel azóta is sajnos már nem követik ezt a jó szokást :(
  - + indulásnál az éppen aktuális koktél részletes receptjét is megkaptuk (cat /etc/banner)

### ▶ 2006 augusztustól

- ▶ build környezet alapvető fejlesztése
- ▶ eredmény: **Kamikaze** első verziója (majd számos kiadása 2010-ig)

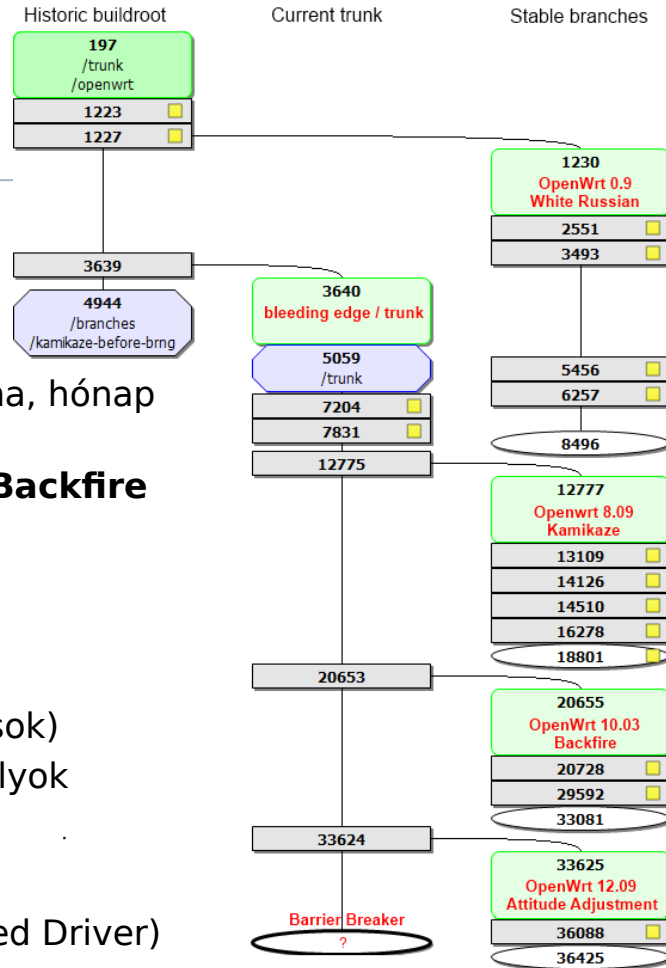


# OpenWRT koktélok

## ► Történet

### ► Verziók

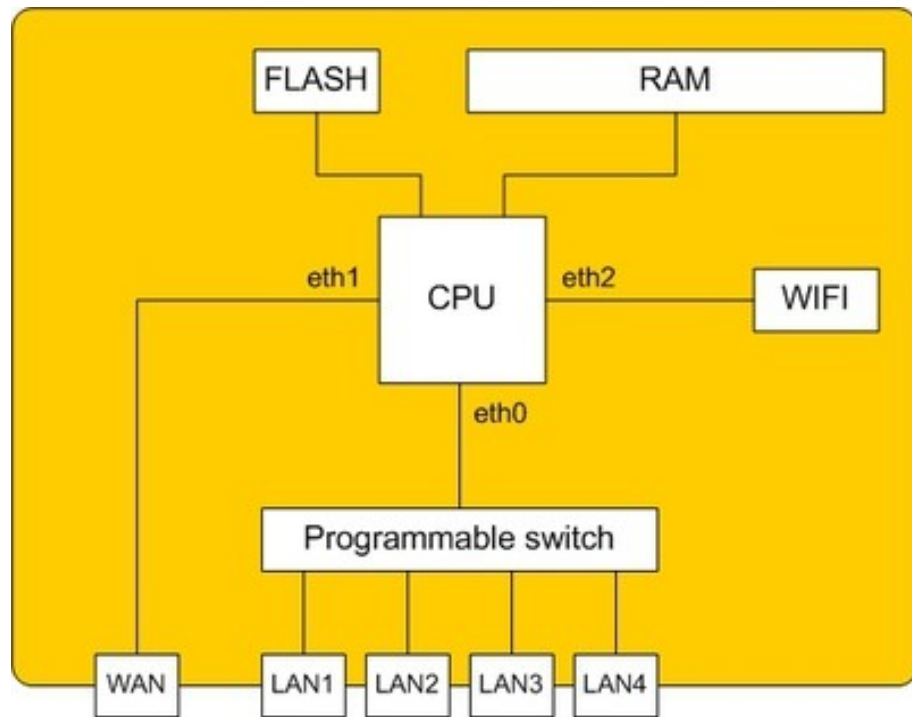
- fő verziószám helyett kódnevek
- ezt egészítik ki különböző számokkal: első kiadás évszáma, hónap [, service release, release candidate sorszám]
- első sokak által használt, sok eszközt támogató verzió a **Backfire** 10.03-as
- 2016-2017: OpenWRT fork: LEDE
  - Linux Embedded Development Environment
  - Oka: nézeteltérés a fejlesztési folyamatokban (valójában kisebbség kezében voltak a központi erőforrások)
  - svn → git, uClibc → musl, demokratikusabb döntési szabályok
- Jelenleg (remerge 2018. jan.)
  - OpenWrt név, LEDE fejlesztési modell
  - Nincsenek kódnevek a release-ben (elmaradt a Designated Driver)
  - “latest stable” verzió: 21.02 (“old stable” verzió: 19.07)



# WiFi routerek felépítése

# WiFi router (egy lehetséges) felépítése

- ▶ Alacsony teljesítményű CPU
  - ▶ hálózati interfészekkel
- ▶ kisebb flash memória perzisztens adatok tárolására
  - ▶ MTD (Memory Technology Device)
  - ▶ speciális “block device”
  - ▶ (raw flash devices)
- ▶ nagyobb RAM
- ▶ programozható switch (chip)
- ▶ WiFi chip
- ▶ egyéb perifériák
  - ▶ pl. USB eszközök
- ▶ soros interfész, JTAG...
  - ▶ ha baj van





# OpenWRT részletek

## Partíciók, fájlrendszerek

# Partíciók, fájlrendszerek

- ▶ MTD speciális eszköz
  - ▶ speciális partícionálás
  - ▶ speciális fájlrendszerek
  - ▶ nincs MBR, PBR
    - ▶ offsetek megadása (lehet címke is)
    - ▶ kernel, bootloader ezek alapján dolgozik
    - ▶ hierarchikus szervezés
- ▶ lekérdezés
  - ▶ `cat /proc/mtd`

TP-Link WR1043ND Flash Layout					
Layer0	m25p80 spi0.0: m25p64 8192KiB				
Layer1	mtd0 <i>u-boot</i> 128KiB	mtd5 <i>firmware</i> 8000KiB			mtd4 <i>art</i> 64KiB
Layer2		mtd1 <i>kernel</i> 1280KiB	mtd2 <i>rootfs</i> 6720KiB		
mountpoint			/		
filesystem			mini_fo		
Layer3			mtd3 <i>rootfs_data</i> 5184KiB		
Size In KiB	128KiB	1280KiB	1536KiB	5184KiB	64KiB
Name	<i>u-boot</i>	<i>kernel</i>		<i>rootfs_data</i>	<i>art</i>
mountpoint	<i>none</i>	<i>none</i>	/rom	/overlay	<i>none</i>
filesystem	<i>none</i>	<i>none</i>	SquashFS	JFFS2	<i>none</i>

# Partíciók, fájlrendszerek

- ▶ Layer0
  - ▶ közvetlenül látszik a fizikai flash chip
- ▶ Layer1
  - ▶ mtd0: bootloader
  - ▶ mtd5: firmware
  - ▶ mtd4: ART (Atheros Radio Test)
- ▶ Layer2
  - ▶ firmware partíciót tovább osztjuk
  - ▶ mtd1: kernel
  - ▶ mtd2: rootfs
- ▶ Layer3
  - ▶ rootfs partíciót szedjük szét két részre:
  - ▶ mtd3: rootfs\_data
  - ▶ név nélküli partíció a fájlrendszer csak olvasható részének (SquashFS)

TP-Link WR1043ND Flash Layout					
Layer0	m25p80 spi0.0: m25p64 8192KiB				
Layer1	mtd0 <i>u-boot</i> 128KiB	mtd5 <i>firmware</i> 8000KiB			mtd4 <i>art</i> 64KiB
Layer2		mtd1 <i>kernel</i> 1280KiB	mtd2 <i>rootfs</i> 6720KiB		
mountpoint			/		
filesystem			mini_fo		
Layer3				mtd3 <i>rootfs_data</i> 5184KiB	
Size In KiB	128KiB	1280KiB	1536KiB	5184KiB	64KiB
Name	<i>u-boot</i>	<i>kernel</i>		<i>rootfs_data</i>	<i>art</i>
mountpoint	<i>none</i>	<i>none</i>	/rom	/overlay	<i>none</i>
filesystem	<i>none</i>	<i>none</i>	SquashFS	JFFS2	<i>none</i>

# Kitekintés: SquashFS és union fájlrendszerek

## ▶ SquashFS

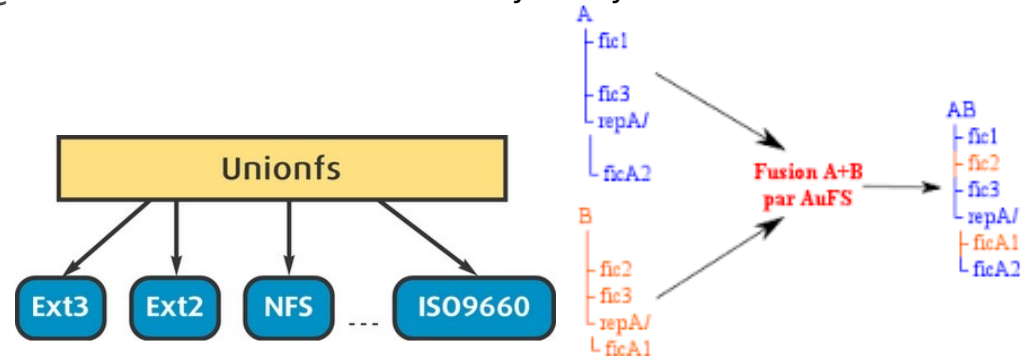
- ▶ tömörített read-only fájlrendszer
- ▶ egy sima fájlban tárolva
  - ▶ egyszerű létrehozás
  - ▶ mksquashfs ...

## ▶ Union fájlrendszerek

- ▶ több fájlrendszer “összefogása” egybe
- ▶ lehetnek ugyanolyan nevű fájlok
  - ▶ prioritást kell definiálni
- ▶ read-only és read-write fájlrendszerek
- ▶ copy-on-write
- ▶ hány réteget támogat?

## ▶ Többféle implementáció

- ▶ unionfs (v1, v2)
- ▶ aufs
  - ▶ nem került be a hivatalos kernelbe
  - ▶ de Docker ezt (is) használja
- ▶ overlaysfs (by Szeredi Miklós)
  - ▶ mainline Linux kernelben
- ▶ mini\_fo
  - ▶ mini fanout overlay file system

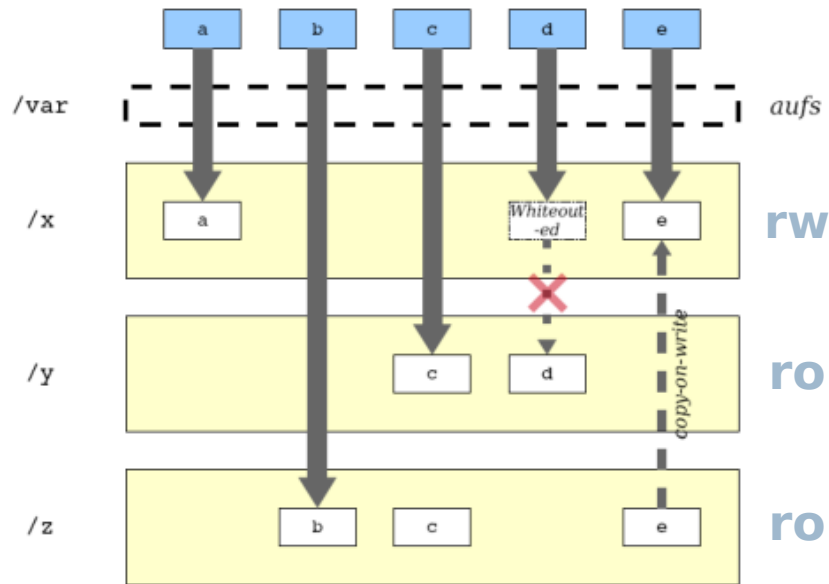


<http://www.linuxjournal.com/article/7714>

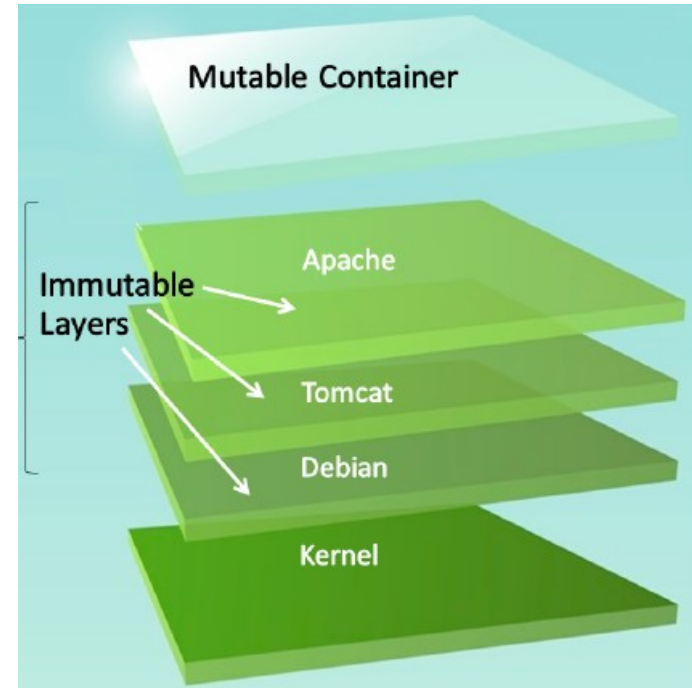
<https://www.admin-linux.fr/aufs-unir-les-contenus-de-repertoires/>

# Kitekintés: SquashFS és union fájlrendszerek

## Példa: ro, rw fájlrendszerek



## Példa: Docker



<http://d.hatena.ne.jp/dayflower/20080714/1216010519>

<https://jaxenter.com/the-resilient-and-highly-scalable-applications-cloud-121731.html>

# Mindez az OpenWRT-ben

---

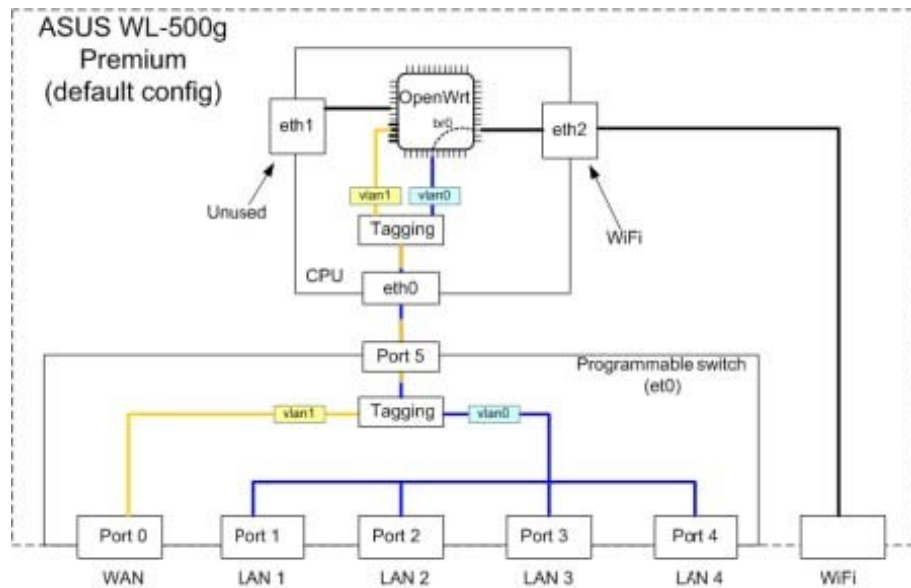
- ▶ fájlrendszer felépítése
  - ▶ /rom
    - ▶ read-only rész (ro)
    - ▶ squashfs
  - ▶ /overlay
    - ▶ írható rész (rw)
    - ▶ JFFS2
    - ▶ (Journalling Flash File System v2)
    - ▶ ide íródnak a különbségek
  - ▶ /
    - ▶ /rom + /overlay
    - ▶ mini\_fo fájlrendszer
    - ▶ ami egyesíti az alatta levő fájlrendszereket
    - ▶ tudja, hogy melyik fájlt hol kell elérni

# OpenWRT részletek

## Hálózatkezelés

# VLAN-ok szerepe

- ▶ Programozható switch (chip)
  - ▶ (jobb esetben) tetszőlegesen konfigurálható
  - ▶ VLAN támogatás
  - ▶ CPU-nak meg kell tudni különböztetni, hogy melyik interfészen jött a csomag
    - ▶ (bizonyos funkciók esetében)
- ▶ megoldás: VLAN
  - ▶ bejövő csomagok VLAN taget kapnak (pl. port száma)
  - ▶ CPU-nál VLAN függő virtuális interfészek (eth0.1, eth0.2, ...)



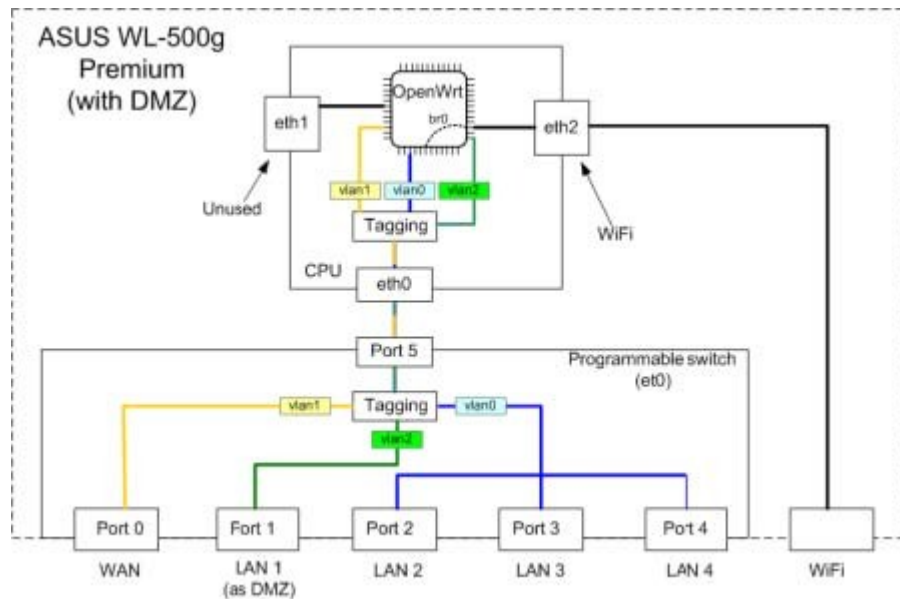
Példa: WAN és LAN portok megkülönböztetése

Forgalom ugyanazon a fizikai interfészen érkezik a CPU-hoz



# VLAN-ok szerepe

- ▶ Programozható switch (chip)
  - ▶ (jobb esetben) tetszőlegesen konfigurálható
  - ▶ VLAN támogatás
  - ▶ CPU-nak meg kell tudni különböztetni, hogy melyik interfészen jött a csomag
    - ▶ (bizonyos funkciók esetében)
- ▶ megoldás: VLAN
  - ▶ bejövő csomagok VLAN taget kapnak (pl. port száma)
  - ▶ CPU-nál VLAN függő virtuális interfészek (eth0.1, eth0.2, ...)



Példa: WAN, DMZ és LAN portok megkülönböztetése

Forgalom ugyanazon a fizikai interfészen érkezik a CPU-hoz

# VLAN-ok szerepe

## ▶ Programozható switch (chip)

### ▶ /etc/config/network

```
config switch
    option name 'switch0'
    option reset '1'
    option enable_vlan '1'

config switch_vlan
    option device 'switch0'
    option vlan '1'
    option vid '1'
    option ports '0t 1 2 3'

config switch_vlan
    option device 'switch0'
    option vlan '2'
    option ports '5 6'
    option vid '2'

config switch_vlan
    option device 'switch0'
    option vlan '3'
    option vid '3'
    option ports '0t 4'

config interface 'LAN4'
    option proto 'static'
    option ifname 'eth1.3'
    option netmask '255.255.255.0'
    option ipaddr '192.168.4.1'

config interface 'lan'
    option force_link '1'
    option type 'bridge'
    option proto 'static'
    option ipaddr '192.168.1.1'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option _orig_ifname 'eth1 wlan0'
    option _orig_bridge 'true'
    option ifname 'eth1.1'
```

## ▶ ifconfig

```
br-lan    Link encap:Ethernet  HWaddr E8:DE:27:F6:F7:AC
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0

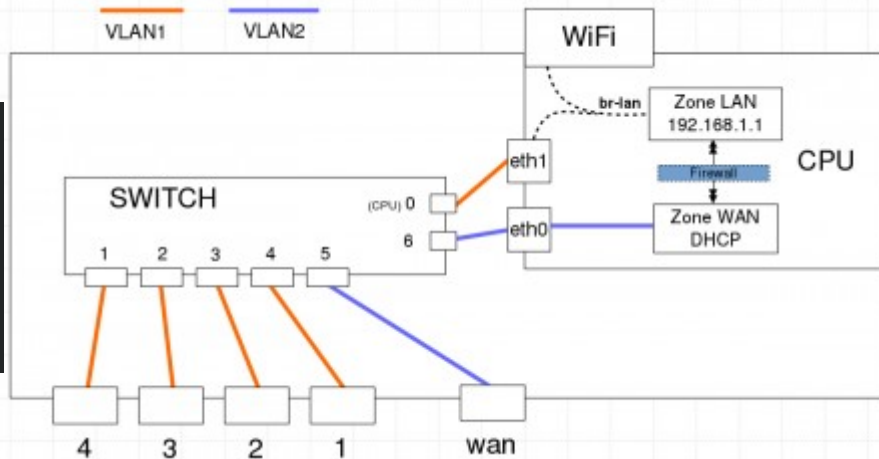
eth1.1   Link encap:Ethernet  HWaddr E8:DE:27:F6:F7:AC
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

eth1.3   Link encap:Ethernet  HWaddr E8:DE:27:F6:F7:AC
          inet addr:192.168.4.1  Bcast:192.168.4.255  Mask:255.255.255.0
```

## ▶ brctl show

bridge name	bridge id	STP enabled	interfaces
br-lan	7fff.e8de27f6f7ac	no	eth1.1 wlan0

TP-Link TL-WR1043ND v2.x, v3.x



# OpenWRT részletek

## Konfiguráció

```
sonkoly@notty:~$ ssh root@192.168.1.1
root@192.168.1.1's password:
```

# Konfigurálás

```
BusyBox v1.22.1 (2014-08-04 22:39:32 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

## ▶ telnet

- ▶ először általában csak ez van
- ▶ password beállítása után: ssh

## ▶ ssh

- ▶ általában egyszerűsített Linux shell (busybox)
- ▶ rugalmas konfiguráció
- ▶ UCI

- ▶ Unified Configuration Interface
- ▶ paraméterek beállítása egységes interfészen
- ▶ konfig fájlok is ezt használják
  - ▶ pl: /etc/config/network
- ▶ csomagkezelés: opkg
  - ▶ ~egyszerűsített dpkg



```
-----
|_| W I R E L E S S   F R E E D O M
-----
BARRIER BREAKER (14.07-rc3, r42056)
-----
* 1/2 oz Galliano           Pour all ingredients into
* 4 oz cold Coffee         an irish coffee mug filled
* 1 1/2 oz Dark Rum        with crushed ice. Stir.
* 2 tsp. Creme de Cacao
-----
root@SobiNet:~# █
```

```
config interface 'lan'
    option force_link '1'
    option type 'bridge'
    option proto 'static'
    option ipaddr '192.168.1.1'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option _orig_ifname 'eth1 wlan0'
    option _orig_bridge 'true'
    option ifname 'eth1.1'
```

# Konfigurálás

---

- ▶ Korábban tanult eszközök használhatók
- ▶ Egy Linux rendszert kell konfigurálni
- ▶ Használhatjuk routerként, switch-ként is
- ▶ Hálózati funkciók (opkg) csomagokban tehetők fel
  - ▶ dhcp, iptables (nat, firewall), dns
  - ▶ egy alap firmware-ben ezek benne vannak
  - ▶ WiFi kezelés (ez jön majd később!)
- ▶ Számos egyéb program
  - ▶ transmission, asterisk, apache2, ...
  - ▶ webcamera kezeléstől az USB-s külső HDD kezeléséig szinte minden
  - ▶ de ha nincs meg: lefordítható forrásból (cross-compilation)

# LuCI: a könnyebb út

## Példa: Network/Interfaces

SobiNet Status System Network Logout AUTO REFRESH ON

### Interfaces

Interface Overview

Network	Status	Actions
<b>LAN</b>  br-lan	<b>Uptime:</b> 250d 11h 22m 56s <b>MAC-Address:</b> E8:DE:27:F6:F7:AC <b>RX:</b> 144.73 GB (868864824 Pkts.) <b>TX:</b> 1.85 TB (1468893468 Pkts.) <b>IPv4:</b> 192.168.1.1/24 <b>IPv6:</b> FDAD:6E0F:efd8:0:0:0:0:1/60	Connect  Stop  Edit  Delete
<b>LAN4</b>  eth1.3	<b>Uptime:</b> 250d 11h 22m 56s <b>MAC-Address:</b> E8:DE:27:F6:F7:AC <b>RX:</b> 5.58 GB (9116791 Pkts.) <b>TX:</b> 6.61 GB (8613095 Pkts.) <b>IPv4:</b> 192.168.4.1/24	Connect  Stop  Edit  Delete
<b>WAN</b>  pppoe-wan	<b>Uptime:</b> 6d 14h 1m 22s <b>RX:</b> 45.37 GB (35980081 Pkts.) <b>TX:</b> 8.51 GB (22538525 Pkts.) <b>IPv4:</b> 188.143.77.204/32	Connect  Stop  Edit  Delete

Add new interface...

### Global network options

IPv6 ULA-Prefix

Powered by LuCI Trunk (svn-r10467) OpenWrt Barrier Breaker 14.07-rc3

# LuCI: a könnyebb út

## Példa: Network/Switch

**SobiNet** Status ▾ System ▾ Network ▾ Logout AUTO REFRESH ON

### Switch

The network ports on this device can be combined to several VLANs in which computers can communicate directly with each other. VLANs are often used to separate different network segments. Often there is by default one Uplink port for a connection to the next greater network like the internet and other ports for local network.

Switch "switch0"

Enable VLAN functionality

#### VLANs on "switch0"

VLAN ID	CPU	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6
Port status:							
	1000baseT full-duplex	1000baseT full-duplex	no link	no link	no link	1000baseT full-duplex	1000baseT full-duplex
<input type="text" value="1"/>	tagged ▾	untagged ▾	untagged ▾	untagged ▾	off ▾	off ▾	off ▾
<input type="text" value="2"/>	off ▾	off ▾	off ▾	off ▾	off ▾	untagged ▾	untagged ▾
<input type="text" value="3"/>	tagged ▾	off ▾	off ▾	off ▾	untagged ▾	off ▾	off ▾

Add

Powered by LuCI Trunk (svn-r10467) OpenWrt Barrier Breaker 14.07-rc3

# LuCI: a könnyebb út

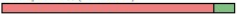
Példa:  
System/Software

SobiNet Status - System - Network - Logout

## Software

Actions Configuration

No package lists available [Update lists](#)

Free space: 9% (404.00 KB) 

Download and install package:  [OK](#)

Filter:  [Find package](#)

### Status

Installed packages Available packages

	Package name	Version
<a href="#">Remove</a>	base-files	155-42056
<a href="#">Remove</a>	block-mount	2014-06-22-e0430f5c62f367e5a8e02755412977b02c3fc45e
<a href="#">Remove</a>	busybox	1.22.1-2
<a href="#">Remove</a>	dnsmasq	2.71-3
<a href="#">Remove</a>	dropbear	2014.63-1
<a href="#">Remove</a>	firewall	2014-07-19
<a href="#">Remove</a>	fstools	2014-06-22-e0430f5c62f367e5a8e02755412977b02c3fc45e
<a href="#">Remove</a>	hostapd-common	2014-06-03-1
<a href="#">Remove</a>	ip6tables	1.4.21-1
<a href="#">Remove</a>	iperf	2.0.5-1
<a href="#">Remove</a>	iptables	1.4.21-1
<a href="#">Remove</a>	iw	3.15-1
<a href="#">Remove</a>	jshn	2014-07-16-bd388d2b6c2c151b513c1e449417d18ce02d10b
<a href="#">Remove</a>	jsonfilter	2014-06-19-cdc760c5807744f40adbbe41e1556a67c1b9a9
<a href="#">Remove</a>	kernel	3.10.49-1-94831e5bcf361d1c03e87a15e152b0e8
<a href="#">Remove</a>	kmod-ath	3.10.49+2014-05-22-1
<a href="#">Remove</a>	kmod-ath9k	3.10.49+2014-05-22-1
<a href="#">Remove</a>	kmod-ath9k-common	3.10.49+2014-05-22-1
<a href="#">Remove</a>	kmod-cfg80211	3.10.49+2014-05-22-1
<a href="#">Remove</a>	kmod-crypto-aes	3.10.49-1
<a href="#">Remove</a>	kmod-crypto-arc4	3.10.49-1
<a href="#">Remove</a>	kmod-crypto-core	3.10.49-1
<a href="#">Remove</a>	kmod-crypto-des	3.10.49-1
<a href="#">Remove</a>	kmod-crypto-ecb	3.10.49-1



# Összefoglalás

---

- ▶ Hálózati eszközök belső felépítése...
- ▶ helyett: OpenWRT
  - ▶ egy jó példa
  - ▶ egy beágyazott Linux (firmware)
  - ▶ pl. WiFi routerek vezérlésére
  - ▶ tetszőlegesen testreszabható, konfigurálható
  - ▶ saját programok írhatók hozzá,
  - ▶ futtathatók rajta
- ▶ **Érdeemes kipróbálni!**