Hálózatok építése és üzemeltetése

OpenWRT

Mai téma

- Hálózati eszközök belső felépítése
  - egy konkrét példán keresztül
- OpenWRT
  - lehet, hogy már inkább LEDE…

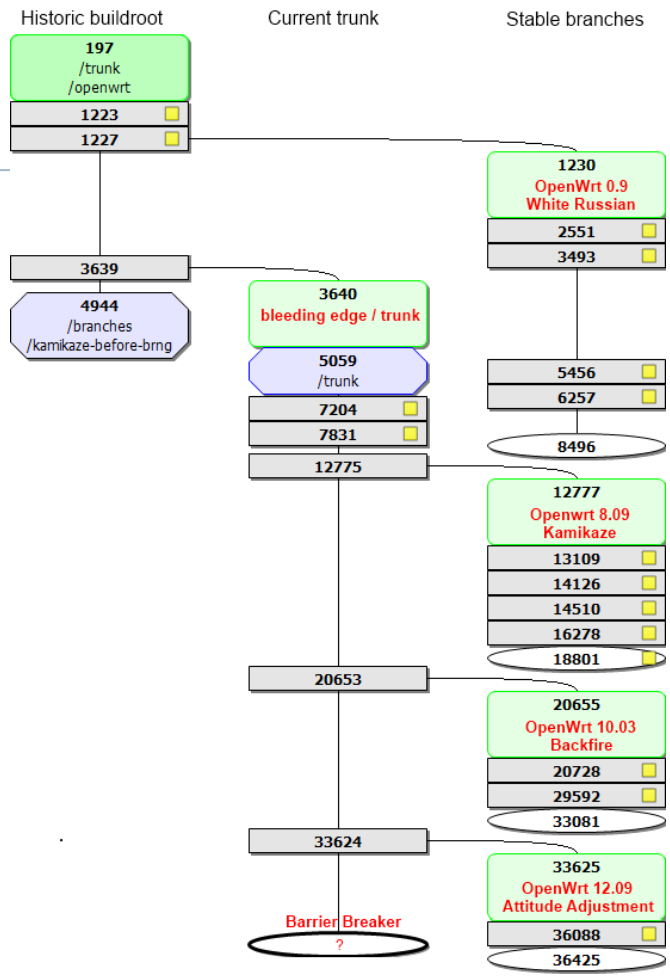
OpenWRT történet

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) tudjunk operációs rendszert / firmware-t “készíteni”
      - funkciók: keresztfordítás (cross-compilation), a root filesystem és a megfelelő kernel generálása, bootloader image létrehozása

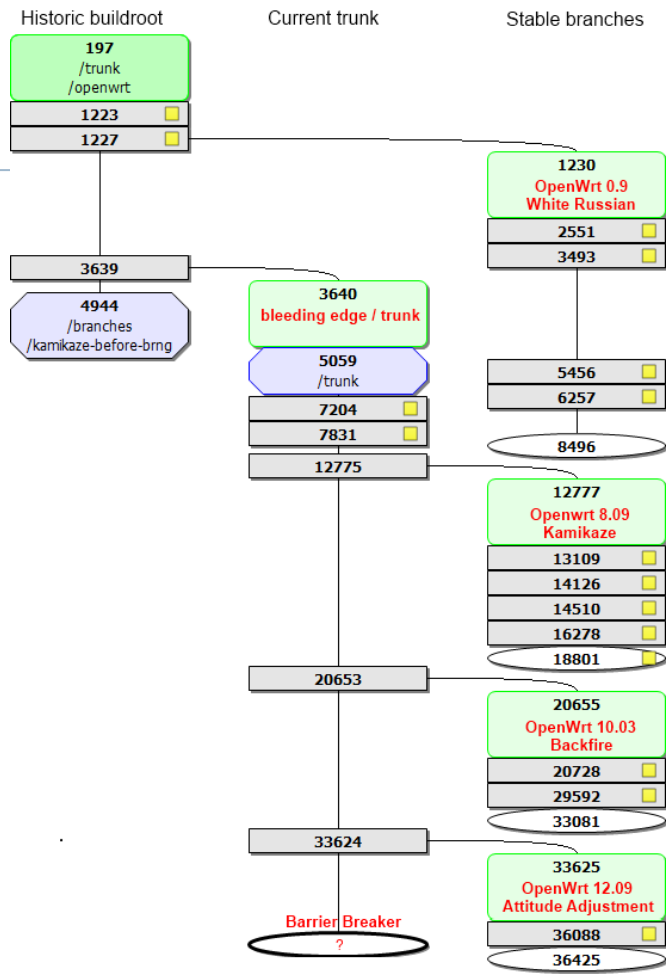
OpenWRT koktélok

- 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 követik ezt a jó szokást
      - + indulásnál az éppen aktuális koktél részletes receptjét is megkapjuk (cat /etc/banner)
  - 2006 augusztustól
    - build környezet alapvető fejlesztése
    - eredmény: **Kamikaze** első verziója (majd számos kiadása 2010-ig)



OpenWRT koktélok

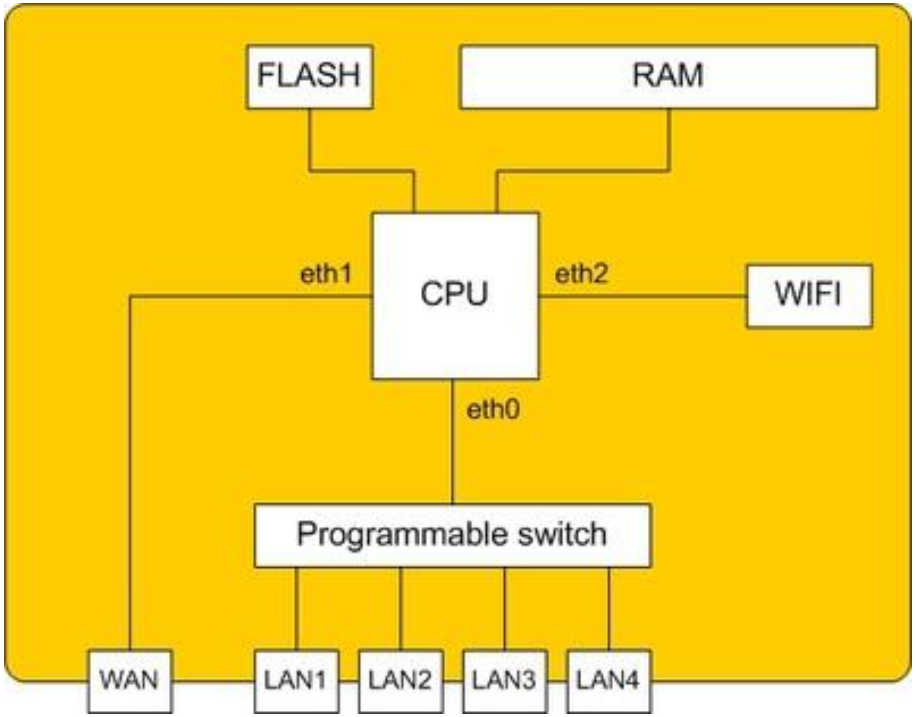
- Történet
  - Verziók
    - fő verziószám helyett kódnevek
    - ezt egészítik ki különböző számokkal: első kiadás évszáma, hónap [, service release, release candidate sorszám]
    - első sokak által használt, sok eszközt támogató verzió a **Backfire** 10.03-as
  - Jelenleg
    - stabil verzió: **Chaos Calmer** 15.05.1
    - development: bleeding edge / trunk
  - Új projekt, OpenWRT fork
    - LEDE
    - Linux Embedded Development Environment



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 cimke 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 ***u-boot*** 128KiB		mtd5 ***firmware*** 8000KiB		mtd4 ***art*** 64KiB
Layer2		mtd1 ***kernel*** 1280KiB	mtd2 ***rootfs*** 6720KiB		
mountpoint	/				
filesystem	mini_fo				
Layer3	mtd3 ***rootfs_data*** 5184KiB				
Size in KiB	128KiB	1280KiB	1536KiB	5184KiB	64KiB
Name	***u-boot***	***kernel***		***rootfs_data***	***art***
mountpoint	*none*	*none*	`/rom`	`/overlay`	*none*
filesystem	*none*	*none*	SquashFS	JFFS2	*none*

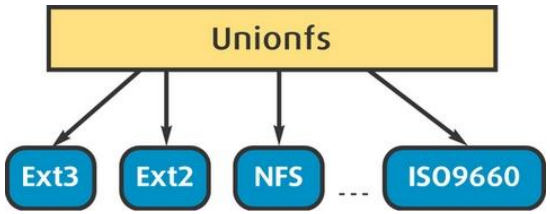
Partíciók, fájlrendszerek

- 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)

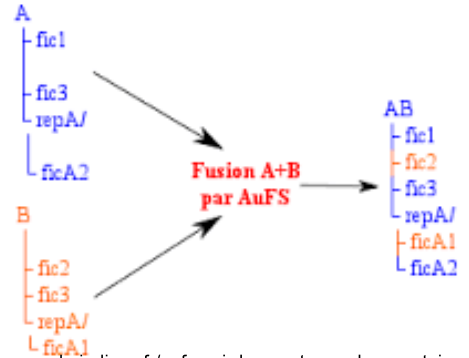
TP-Link WR1043ND Flash Layout					
Layer0	m25p80 spi0.0: m25p64 8192KiB				
Layer1	mtd0 ***u-boot*** 128KiB		mtd5 ***firmware*** 8000KiB		mtd4 ***art*** 64KiB
Layer2		mtd1 ***kernel*** 1280KiB	mtd2 ***rootfs*** 6720KiB		
mountpoint			/		
filesystem	mini_fo				
Layer3			mtd3 ***rootfs_data*** 5184KiB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

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
  - overlayfs (by Szeredi Miklós)
    - mainline Linux kernelben
  - mini_fo
    - mini fanout overlay file system



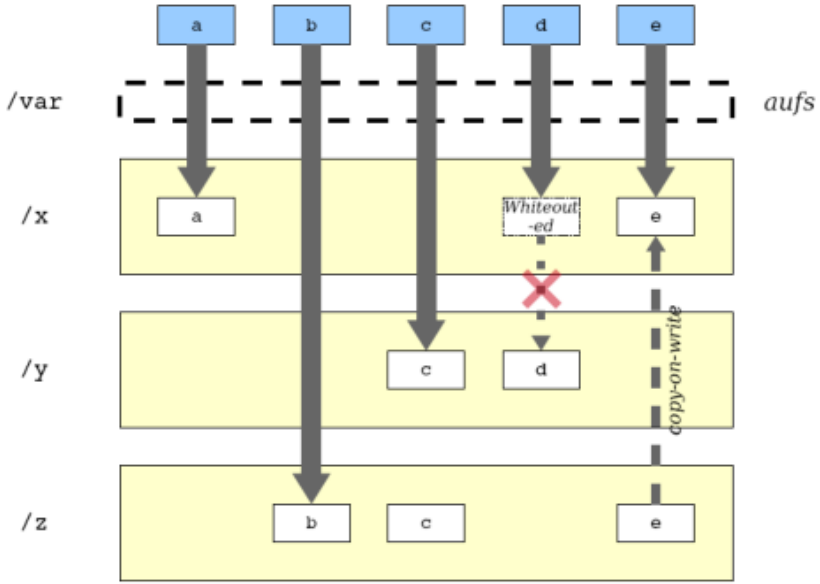
http://www.linuxjournal.com/article/7714



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

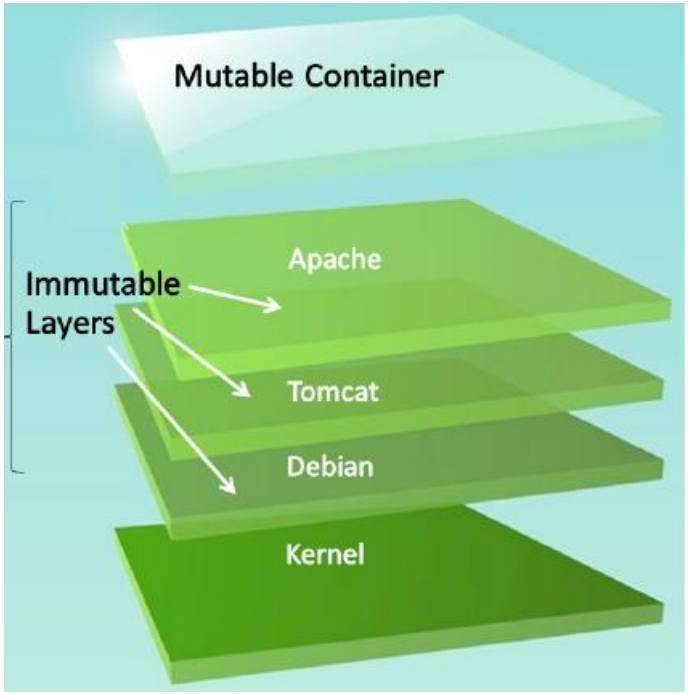
Kitekintés: SquashFS és union fájlrendszerek

Példa: ro, rw fájlrendszerek



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

Példa: Docker



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

Mindez az OpenWRT-ben

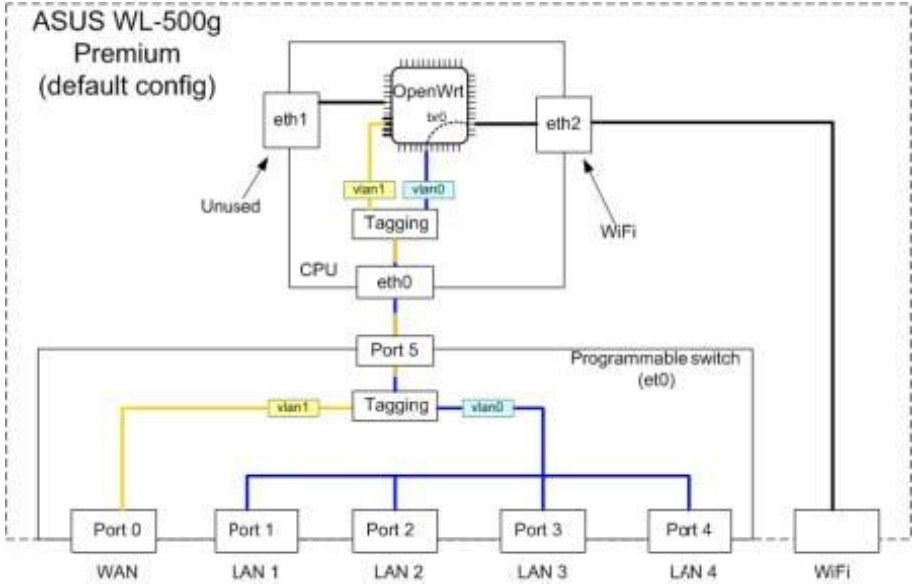
- fájlrendszer felépítése
  - /rom
    - read-only rész (ro)
    - squashfs
  - /overlay
    - írható rész (rw)
    - JFFS2
    - (Journalling Flash File System v2)
    - ide íródnak a különbségek
  - /
    - /rom + /overlay
    - mini_fo fájlrendszer
    - ami egyesíti az alatta levő fájlrendszereket
    - 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, …)

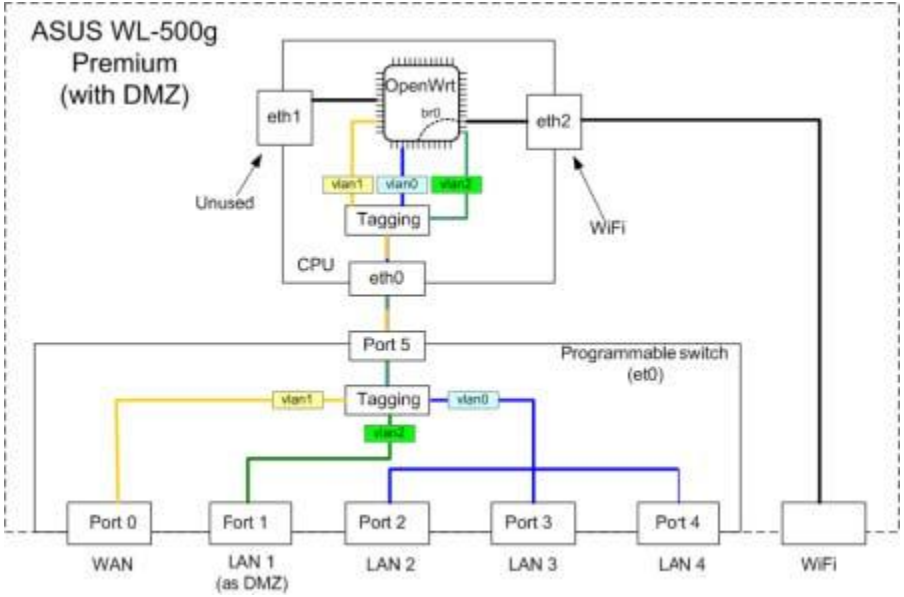


Példa: WAN és LAN portok megkülönböztetése

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

OpenWRT részletek

Konfiguráció

Konfigurálás

- 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

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


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

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| 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:~#
```

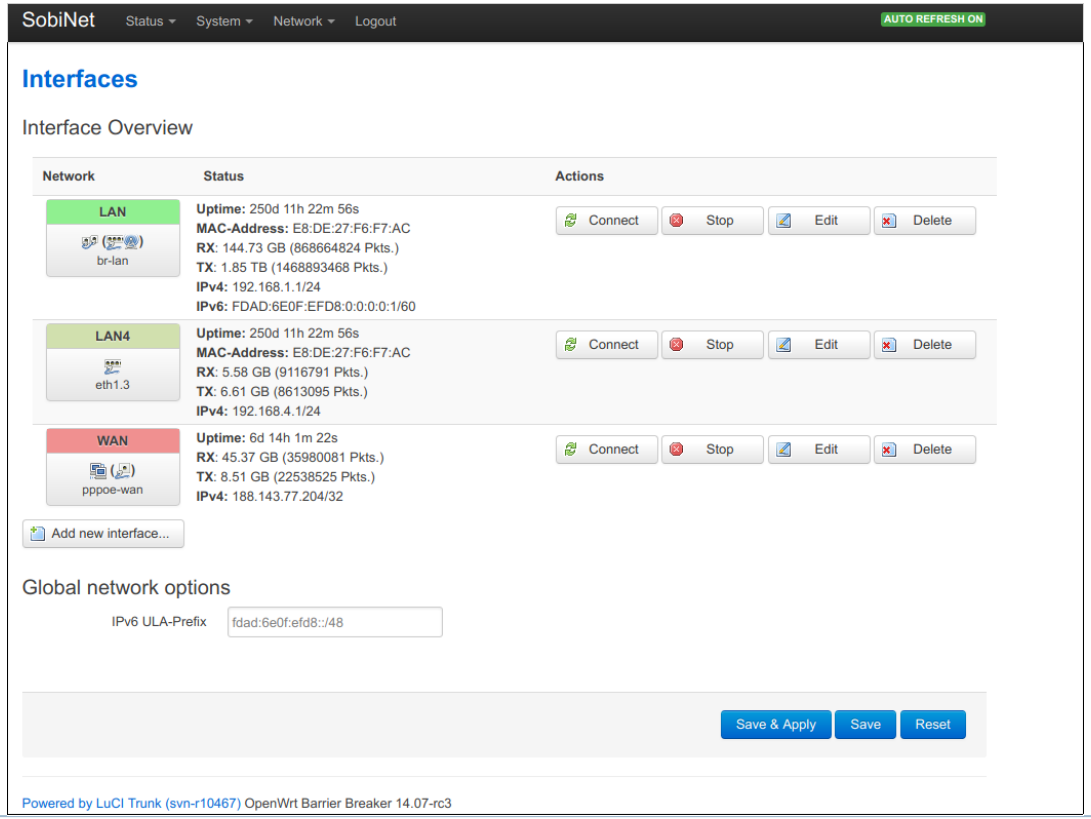
```
config interface 'lan'
        option force_link '1'
        option type 'bridge'
        option proto 'static'
        option ipaddr '192.168.1.1'
        option netmask '255.255.255.0'
        option ip6assign '60'
        option _orig_ifname 'eth1 wlan0'
        option _orig_bridge 'true'
   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

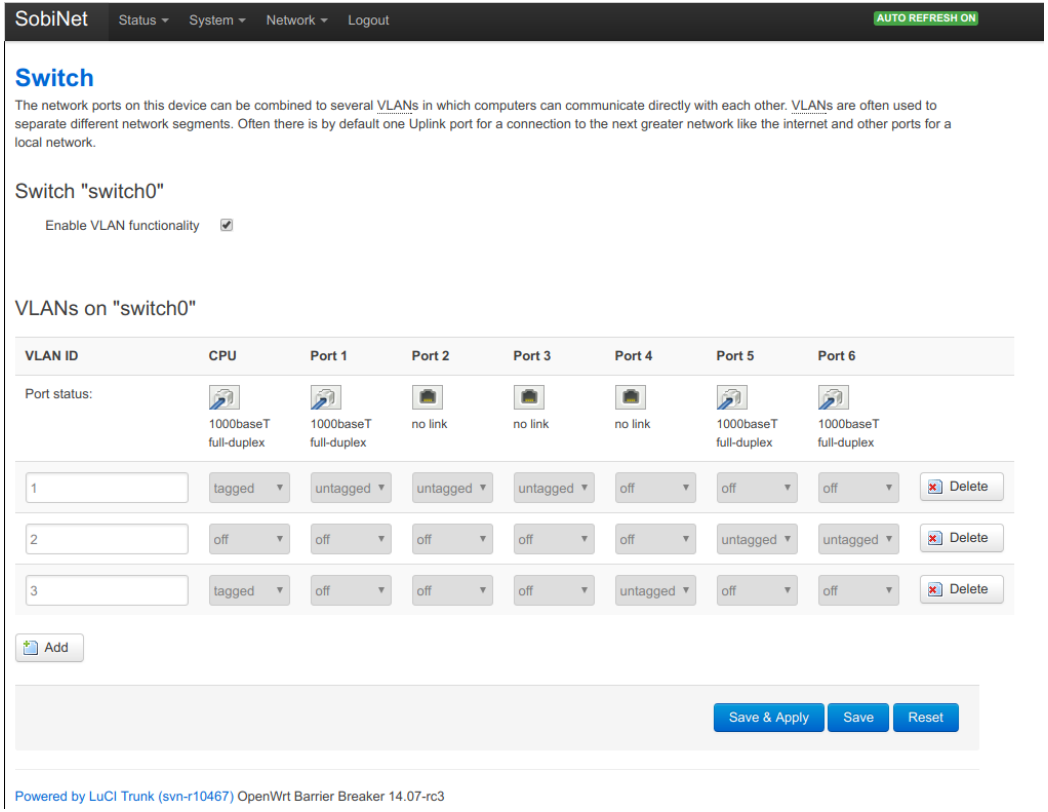
LuCI: a könnyebb út

Példa:
Network/Interfaces



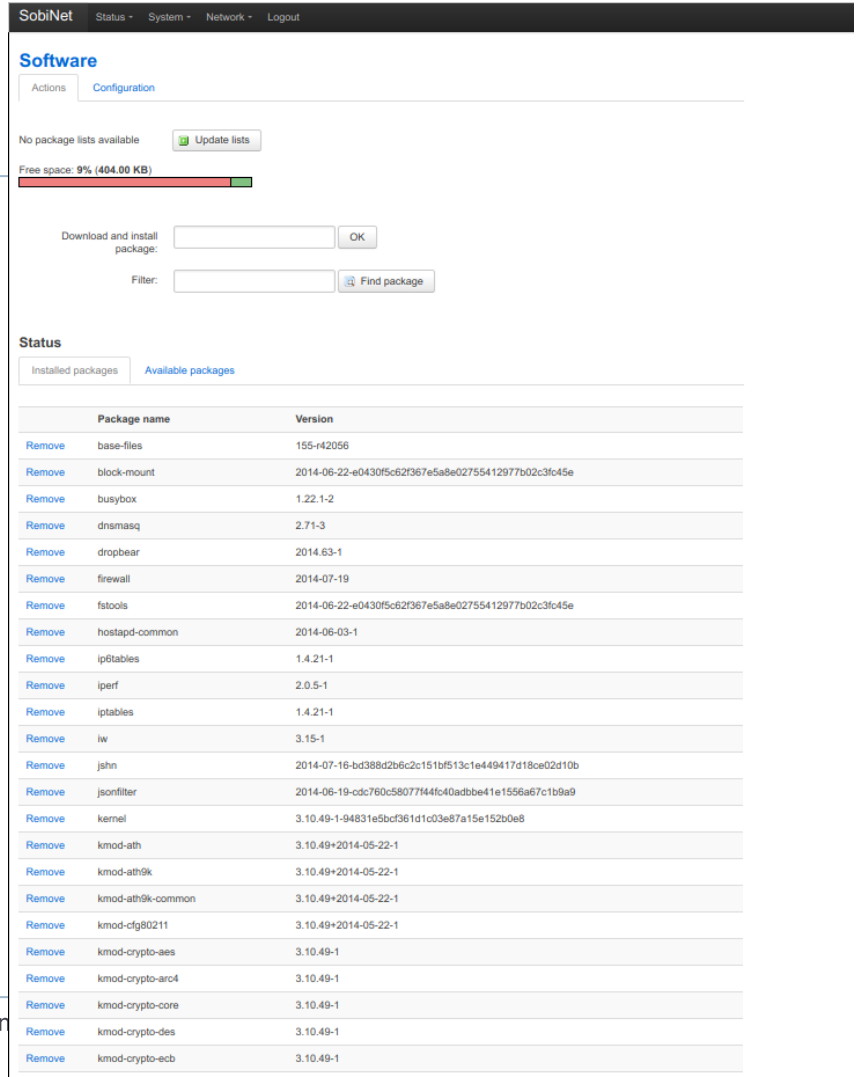
LuCI: a könnyebb út

Példa:
Network/Switch



LuCI: a könnyebb út

Példa:
System/Software



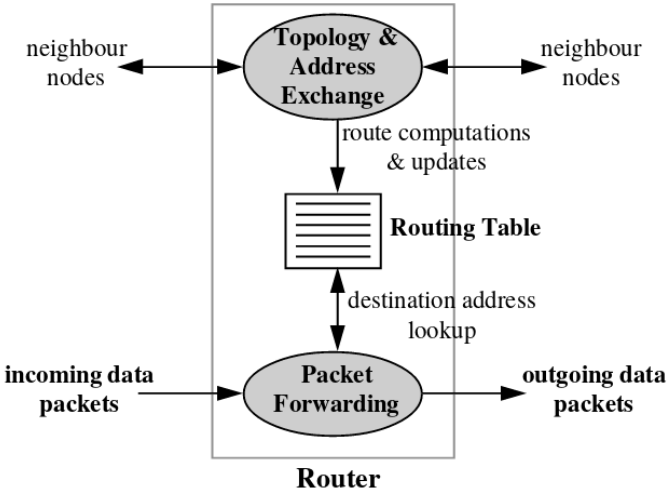
Összefoglalás

- 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
- Érdemes kipróbálni!

(nem otthoni) routerek

home, access, **edge, core**

Routerek általános felépítése

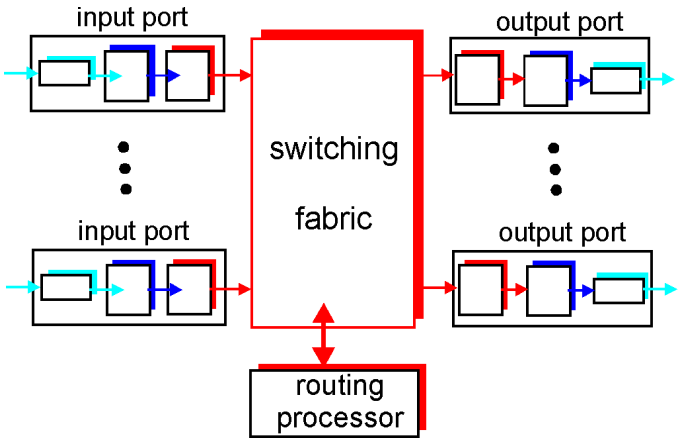


James Aweya: IP Router Architectures: An Overview

A router feladatai

- Routing feladatok
  - routing protokoll (OSPF, …) futtatása, a kapott információk alapján:
  - útvonalak kiszámítása, amit felhasználva:
  - a továbbítási táblázat karbantartása
- Csomagtovábbítási feladatok
  - IP csomag validálása
  - Célcím kikeresése a továbbítási táblázatból
    - helyi, unicast, multicast cím
  - Csomag-élettartam szabályozás
    - TTL, Time-to-live mező csökkentése
  - Ellenőrzőösszeg újraszámítás
  - Eltérő MTU esetén csomagdarabolás

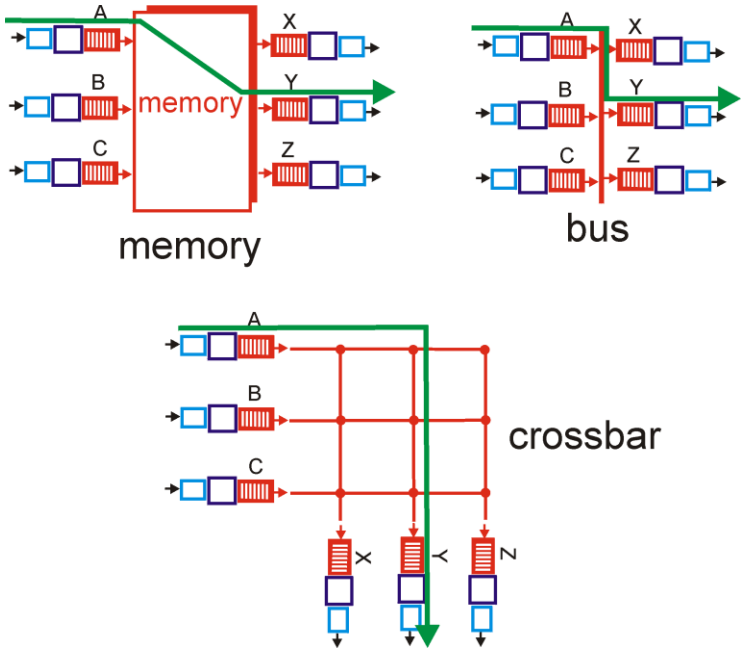
Router architektúrák



http://www2.ic.uff.br/~michael/kr1999/4-network/4_06-inside.htm

- A portok “line card”-okon csatlakoztathatók
- A továbbításhoz szükséges legtöbb feladatot a line cardokon el lehet végezni
- Ideális esetben a routing processor nem végez csomagtovábbítási feladatokat
  - A line cardok rendelkeznek routing tábla másolataival

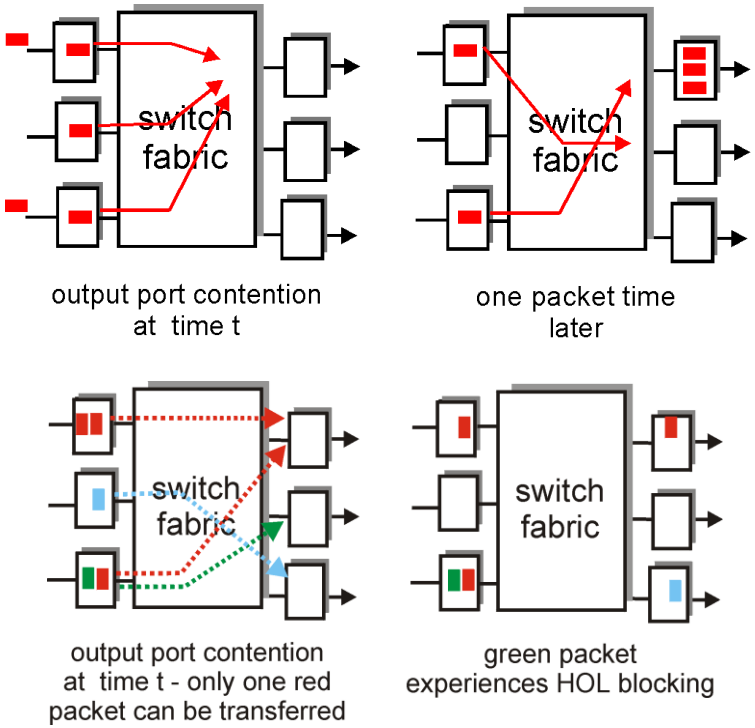
Switching fabric / kapcsolószerkezet



- Kezdetben hagyományos számítógép architektúrákat használtak (~ OpenWrt)
  - csomagokat a közös hozzáférésű memóriába másolták
  - az egyetlen CPU döntött a sorsáról
- 2nd gen: csomagot a közös buszon továbbították a routing processor beavatkozása nélkül
- Crossbar: 2N busz, N input-output port összekapcsolására
  - (szokás ilyenkor a csomagot kisebb egységekre darabolni, majd az output porton összerakni)

Sorbanállás, sormenedzsment, QoS

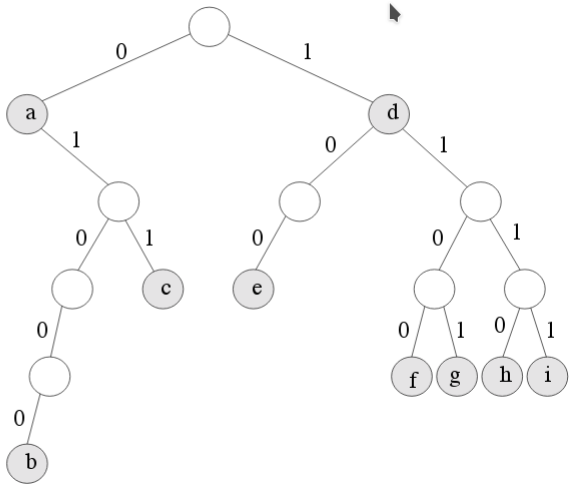
- Sorok a be- és kimentet is lehetnek
  - Kimeneti torlódás oka: pl. nagyobb a bejövő összterhelés a kimeneti kapacitásnál
  - Head-of-the-line blocking:
    - a switching fabric nem N-szeres sebességgel üzemel,
    - a pirosat nem lehet még továbbítani, a zöldet lehetne, de a piros feltartja
- Minőség biztosítása (Quality of Service, QoS)
  - Nem FIFO sorokkal, pl:
  - Prioritásos sorokkal,
  - Weighted Fair Queueing
  - Random Early Detection (RED)
    - Torlódás jelzése csomagdobás nélkül (ECN bitben)
    - TCP estén működik



Kimenő port kiválasztása: Longest Prefix Match

- BGP ~700k bejegyzés
- Egyszerű bináris keresés
  - Túl sok memória hozzáférés
- TCAM (ternary content-addressable memory)
  - Túl drága a teljes táblázat tárolásához
- Gyorsítási lehetőségek:
  - A fa tetejét direktben címezzük
  - Fából irányított gráfot készítünk
    - Pl. a alulról összevonjuk az azonos részfákat
    - Kisebb fa befér a cache-be → gyorsabb a memóriaelérés
  - …

Prefixes
a 0*
b 01000*
c 011*
d 1*
e 100*
f 1100*
g 1101*
h 1110*
i 1111*



Martin Heusse: Router and switch architecture