Update from Turris OS 3.x to 5.x or newer is a big leap
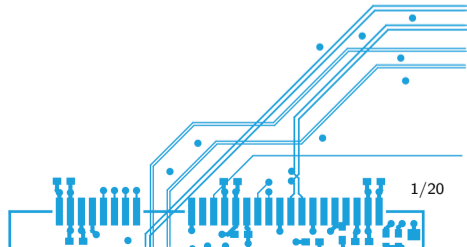
Karel Kočí
29.11.2021

Turris OS 3.x migration for users

Documentation: https://docs.turris.cz/geek/tos3-migration/

Package lists

> **Migration to Turris OS 5.x**
> ☑ This is migration to the latest version of Turris OS. Make sure that you thoroughly read
> https://docs.turris.cz/geek/tos3_migration/ before enabling this!

Automatic migration is going to be in waves. The first wave for:

- ▶ Omnia with serial numbers ending with zero in decimal format
- ▶ Turris 1.x routers on BTRFS
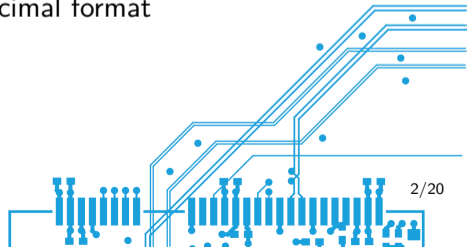
Important changes

- ▶ API tokens are no longer available and mobile application no longer works.
- ▶ Server side backups were dropped. Replacement is not yet provided.
- ▶ Majordomo is no longer available as well as its package list.
- ▶ Automation package list with Domoticz and Home-Assistant is no longer provided.
- ▶ Print server CUPS was dropped. Print server package list now provides only p910nd protocol.
- ▶ Webcam, sound card, Squid proxy and netutils packages list were dropped.
- ▶ Thermometer package was dropped. Use collectd or sensors as a replacement
- ▶ Network switch configuration now uses DSA.
- ▶ Attack statistics are presented on the Sentinel View instead of project.turris.cz.
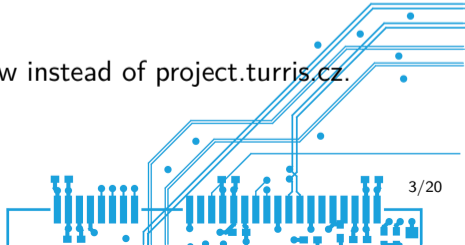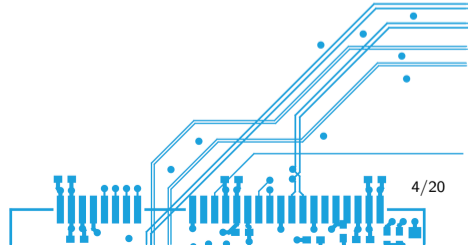
Why we need migration?

Huge changes between years 2016 and 2018.
Turris OS 3.x: OpenWrt 15.05 (latest merge January 2016)
Turris OS 4.0: OpenWrt 18.06

- ▶ Huge changes in configuration
- ▶ Change of LibC on Turris 1.x
- ▶ Changes in repositories (different layout)
- ▶ Changes in packages
- ▶ Removal of some functionalities

Configuration

- Migration from swconfig to DSA pro for switch configuration
- SFP support in kernel (missing switch in runtime)
- Change of Wi-Fi cards identifier
- Move away from UCI configuration to native for sshd
- Update LXC from 1.0 to 3.0 (4.0)
- Switch from uCollect to Sentinel

Load of changes in secondary software..
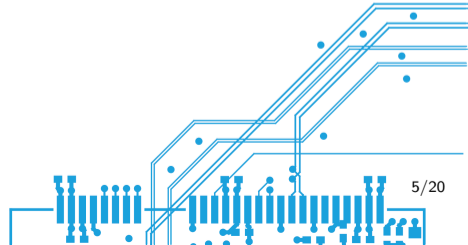
Configuration migration

Package *tos3to4* with migration scripts in Turris OS 4.0+ due to future extension.

```
config_load wireless
wifi_handle() {
  local radio="$1"
  config_get path "$radio" path
  mac="$(cat /sys/devices/platform/"$path"/ieee80211/phy*/macaddress)"
  uci set "wireless.$radio.macaddr=$mac"
  uci delete "wireless.$radio.path"
}
config_foreach wifi_handle wifi-device
uci commit wireless
```
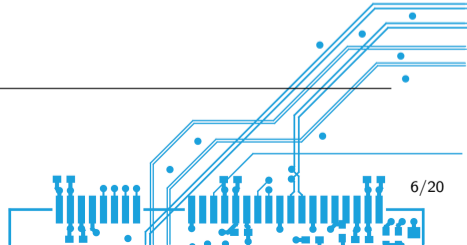
Offline upgrade

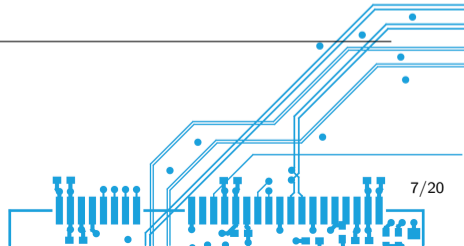Load the new system, migrate configuration and perform reboot...

Advantages:

- ▶ Simple
- ▶ Atomic thanks to BTRFS

Why not possible:

```
[ -n "$IPKG_INSTROOT" ] || {
  package_migration
}
```
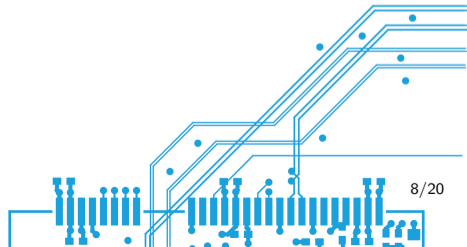
⇒ Upgrade in runtime

Some theory... Updater

Lua directives/configuration from online source (updater lists).

```
Repository('turris/core', '../packages/core')
Install('updater-ng', 'updater-supervisor', { critical = true })
Package('updater-ng', { replan = 'finished' })
```

Some theory... Package extras

```
Package('updater-ng', { replan = 'finished' })
```

▶ **virtual**: there is no binary package for this name
▶ **deps**: additional package dependencies
▶ **reboot**: performs *delayed* reboot or unconditional one for *finished*
▶ **replan**: *finished* causes reexecution of updater immediately after exit or immediately after installation of that package for *immediate*
▶ **abi_change** and **abichange_deep** causes reinstall of all dependent packages
▶ **order_before** and **order_after** never worked
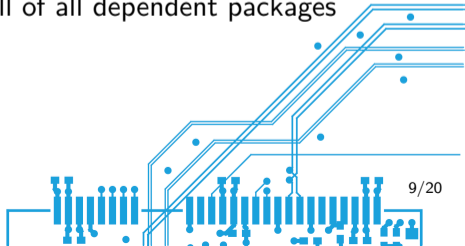
First iteration: "this won't work"

Package *tos3to4* in Turris OS 3.x:

```
Script("https://repo.turris.cz/hbs/omnia/lists/migrate3x.lua", {
        security = "Local", ca = system_cas, crl = no_crl,
        pubkey = { "file:///etc/updater/keys/release.pub" }
})
```

migrate3x.lua on server:

```
Repository("turris", "https://repo.turris.cz/hbs/"..board.."/packages", {
        priority = 60,
        subdirs = { "base", "core", "packages", "turrispackages"}
})
```

Second iteration: "update updater"

```
if version_match(self_version, "<64.0") then
  Repository("turris", "https://repo.turris.cz/hbs/"..board.."/packages",
    priority = 60,
    subdirs = { "base", "core", "packages", "turrispackages"}
  })
  Install('updater-ng', { critical = true })
  Package('updater-ng', {
    replan = 'immediate',
    deps = { 'libgcc', 'busybox', 'tos3to4-early' }
  })
end
Package("libc", { abi_change_deep = true })
```
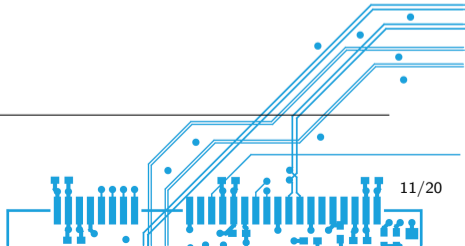
Updater: welcome in hell

Check for collisions...

```
DIE: [string "transaction"]:334: [string "transaction"]:158: Collisions:
• /usr/sbin/cron: vixie-cron (existing-file), cronie (new-file)
```

Ok, then:

```
Package("cronie", { virtual = true })
```

```
ERROR: inconsistent: Candidate exists for virtual package gettext-tools
```

Theory follow up... order of changes

```
Install('updater-ng', { critical = true, priority = 50 })
```

Ordering rules (higher means earlier):

1. Dependent for any subsequent package
2. Package conflicting with some of the subsequent packages
3. Critically requested packages
4. Priority (higher means earlier)
5. Package installation (package removal are performed at the end)

BTW, this applies just for newer versions of updater.
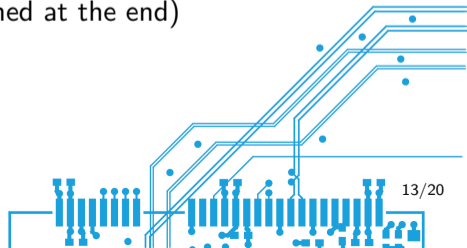
Third iteration "early later"

```
Package('updater-ng', {
  replan = 'immediate',
  deps = { 'libgcc', 'busybox' }
})
else
  if version_match(installed["tos3to4"].version, "<2.0.0") then
    Install('tos3to4-early', { critical = true, priority = 100 })
    Package('tos3to4-early', { replan = 'immediate' })
  end
end
```
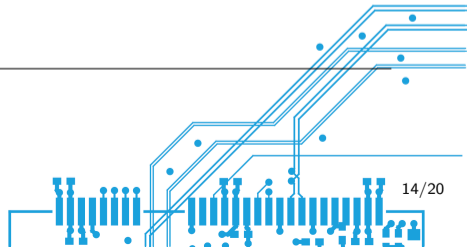
Fourth iteration "ufff it takes sooo long"

OpenWrt uses procd as the init system which communicates over ubus.
New version of ubus does not communicate with the old one. The timeout is two
minutes.

```
[ -n "${IPKG_INSTROOT}" ] || {
  ubus call system info 2>/dev/null >&2 || killall ubusd
}
```

```
if version_match(installed["ubus"].version, "<2018") then
  Package("ubus", { replan = 'immediate' })
end
```
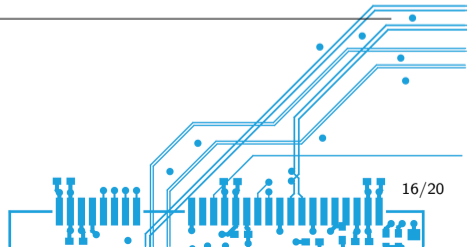
Fifth iteration "missing packages"

```
Install("cups")
```

```
ERROR: inconsistent: Requested package cups that is not available.
```

With new feature added to updater:

```
Mode("optional_installs")
```

Sixth iteration "localrepo oh localrepo"

Localrepo stores packages that were not installed from repository.
Those package won't be updated and won't most likely work after upgrade.

```
SOURCE="/usr/share/updater/localrepo"
COLLIDED="/usr/share/updater/collided$SOURCE"
mv "$SOURCE" "$COLLIDED"
```
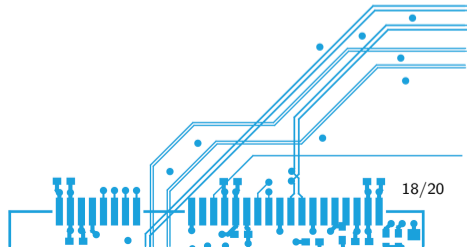
```
  Package('updater-ng', {
    replan = 'immediate',
    deps = { 'libgcc', 'busybox', 'tos3to4-earliest' }
  })
else
```
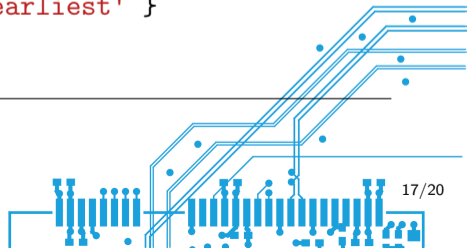
Seventh iteration "brick"

The system is configured for DSA but there is no DSA in running kernel.

```
Package("tos3to4", { reboot = "finished" })
```

Eighth iteration... well no, that would be too long

The complete update consists of:

1. Installation of *tos3to4* in Turris OS 3.x
2. Upgrade of updater to the newest version and installation of *tos3to4-earliest*
3. Upgrade of ubus (two minutes of waiting)
4. Installation of *tos3to4-early* and updater settings migration
5. Upgrade of the rest of the system including removal of old packages
6. The last replan
7. Reboot
8. Removal of *tos3to4* a other migration packages

Plus few other replans in the meantime and special packages (*fix-**).
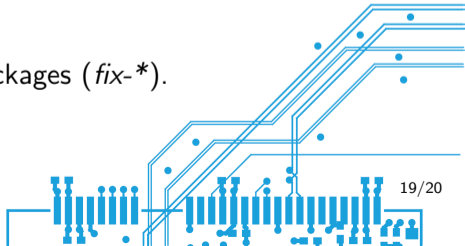
Thank you for attention. Questions?

karel.koci@nic.cz