

Quo vadis linux networking

Pavel Šimerda
pavlix@pavlix.net

LinuxAlt 2012, Brno

<http://data.pavlix.net/linuxalt/2012/>

The story of network management

- Kernel configures the loopback and link-local addresses
- The `ip` command can tweak network interface configuration
- Network configuration is saved in a shell script

Kernel and iproute2 – use cases

- Suitable for testing networking features
- Usable for simple and static environments

Script-based static networking

- A tiny layer over iproute2 or ifconfig
- Various configuration file formats
- Possibility to activate/deactivate individual connections

Script-based static networking – use cases

- Suitable only for manual (aka static) addressing
- Usable for single-network dynamic IPv4 configuration
- Unusable for dynamic IPv6 configuration
- Unusable for multiple dynamically configured networks
- Bad interoperability with other tools including VPNs and DNSSEC

Script-based static networking – examples

- ifcfg variants (Fedora, openSUSE, Mandriva, ...)
- ifupdown (Debian, Ubuntu, ...)
- ifnet (Gentoo, ...)
- uci network (OpenWRT)
- ...

Network configuration daemon

- Provides a central coordination point
- Can aggregate much of the configuration
- Communicates with other tools over IPC

Network configuration daemon – use cases

- Multiple managed network interfaces
- Static & dynamic IPv4 and IPv6 address configuration
- Device prioritization for routing
- Event-based coordination of networking tools
- API for user configuration tools

Network configuration daemon – everybody wants one

- NetworkManager (used by most distributions)
- connman (came from Intel's Meego project)
- Wicd (a network daemon written in Python)
- netcfg (Archlinux)
- netifd (new OpenWRT development)
- wicked (is not Wicd)
- ...?

NetworkManager, distributions' daemon of choice

- Stable branch 0.9.6, development branch 0.9.7
- Long history of laptop and desktop usage
- Actively developed project with active community
- Influences kernel, wpa_supplicant and other projects
- May also influence IETF standardization
- Users who avoid NM still benefit from our work

- Was not designed for servers and virtualization
- Has a history of bugs and regressions
- Relies on various tools with various problems

Nothing is perfect. . .

- Netlink-route was always somewhat unreliable
- IPv4 addresses lack DHCP flag and lifetime
- IPv6 route preference is ignored
- RA routing information is only exposed through routing table
- TCP with temporary addresses breaks on readdressing
- Kernel drops DNS information received through RA
- Bridge loses MAC address when enslaving first device

- `rtnl_link_bond_add()` succeeds when device already exists
- `rtnl_link_brindge_add()` and `rtnl_link_vlan_add()` is missing
- Some distributions still lack libnl 3.x

- IPv4 and IPv6 localhost break (with AI_ADDRCONFIG)
- IPv6 link-local addresses break (with AI_ADDRCONFIG)
- GLIBC lacks support for `getaddrinfo()` in nsswitch
- Implementing link-local multicast DNS is therefore impossible
- It would be nice if `getaddrinfo()` supported SRV records

- `ip` doesn't show interface types
- `ip` doesn't show VLAN IDs
- `ping` still can't handle IPv6 addresses (`ping6` is required)

- We maintain any problems in the NetworkManager bugzilla
- Connection dependency problems (autoconnect and reconnect)
- IPv6 dynamic reconfiguration is not currently supported
- DNSSEC support is not even properly designed, yet
- NetworkManager is currently not testable
- Many important features are missing, broken or undocumented

Current development of NetworkManager

- Desktop and laptop (we're not abandoning you!)
- Simple server (especially command-line interface)
- Virtualization host (the most complicated)
- Network-enabled initramfs (e.g. for NFS or iSCSI boot)
- Simple routing use cases (e.g. connection sharing)
- Controlled environment (NM used as networking API)

Platform interaction redesign

- The whole platform interaction code is being rewritten
- Hacks for kernel bugs will be in one place
- Compatibility hacks for libnl versions likewise
- Platform code is designed to be testable
- Fake platform code enables testing of NM core
- Bridging, bonding and VLAN are built on top

Configuration model redesign

- Separate runtime and persistent configuration
- Saving, restoring and modification APIs
- Generic device configuration
- Accepting modifications from other tools
- Taking over connections on startup and at runtime

Flexible distribution support

- Removed `--with-distro` option, builds for for any distribution
- Added `--enable-netconfig` (SUSE's alternative to `--enable-resolvconf`)
- Plugins are enabled separately (e.g. `--enable-ifcfg-rh`)
- Initscripts are now not installed automatically
- Loopback device is now handled in a generic way
- Other distribution-specific code was removed

DHCP support improvements (dhclient)

- Searching for configuration at runtime
- DHCPv6 support equivalent to DHCPv4
- Dropping support for IPv4-only dhclient 3.x

Introduction of gsystem

- Frontend to `__attribute__((cleanup))`
- Automatic resource cleanups
- Dependency on supporting compilers (GCC, Clang)
- Already used by some other projects
- Not acceptable for Glib, because of MSVC

For the linux networking community

Work with us to make linux networking better

- Discuss at #nm at Freenode or on the mailing list
- File and comment bug reports in our bugzilla
- Improve any projects we rely on and send patches
- Reduce excessive work duplication

<http://data.pavlix.net/linuxalt2012/>

psimerda@redhat.com
pavlix@pavlix.net