Switch Software and Testing

Alessandro Rubini
Independent Consultant
Contract Professor, University of Pavia
rubini@linux.it  rubini@gnudd.com  rubini@unipv.it

GSI, Darmstadt, 2001-04-14
Status of the WRS Package (2011-04-14)

There is a WR sub-project in ohwr.org
- The sub-project is mainly a separate repository
  - The code is currently hosted in my own system
  - Splendoeo is finalizing git on ohwr.org there days

Contents of the wr-switch-sw repository
- Initially, the repo was just for the NIC driver
- Kernel patches for 2.6.35 are hosted there as well
  - I have my own git for kernel, but it’s overkill
  - Should we consider hosting the kernel on ohwr/github ?
- It now includes the RTU kernel driver (Miguel, Juan Luis)
- Over time, the plan is stealing all software from WR project
  - I’ll go on picking people work when git is on ohwr.org
  - This requires some serious patience from the other parties

My effort is in cleaning up and documenting as stuff gets in
- Code cleanup is useful form maintainability
- Real hackers will rate much better clean and documented code
- Documentation is needed for new users to get fast into the project
Other repos, to be folded in the sw repo (?)

**Build Scripts** - includes 8-pages manual, in sync with code
- The package is a clean-up of Tom's work
- It includes complete documentation for ease customization
- Currently it includes basic rootfs, boot, kernel

**minipc** (mini-ipc -- a misnomer for mini-rpc)
- My power grab over Tom's work
  - More portable (no assembly), cleaner, documented
- Is it really worth the effort?

**wrs-testing** - includes 9-pages manual, ahead of code
- Meant to include standard-compliant testing code
  - Bandwidth, frame-rate, latency
- It currently hosts sample code for hw timestamping
  - It includes a patch to fake hwtstamp in RTL8169
  - A similar patch for RTL8139 is being worked on, for qemu

**swflood**
- Switch-flood -- aka octopus
switch flood (the octopus): current status

The tool sends and receives frames on several eth ports
  • The initial idea was just to test the WRS RTU subsystem
    ◦ I bought a few copper-sfp to do that, but they’re still unused

Extra goodies
  • "multidump", sniffing on several ports at the same time
  • "netspit", building raw packets from cmdline arguments

Actions are driven by a config file
  • The commands tells what to send from which macaddr
  • The commands tell what to expect back from the ports

Example: size of the MAC table in a switch
  testing from 8192 different senders plus another
cfg-test/tablesizetest:60: expect failed
expected: 8192 0 0 0
received: 8192 0 2 2
testing from 16384 different senders plus another
cfg-test/tablesizetest:66: expect failed
expected: 16384 0 0 0
received: 16384 0 8194 8194