freeRouter.nop.hu

Networking swiss army knife

Csaba Mate
NIIF/Hungarnet
as1955

About

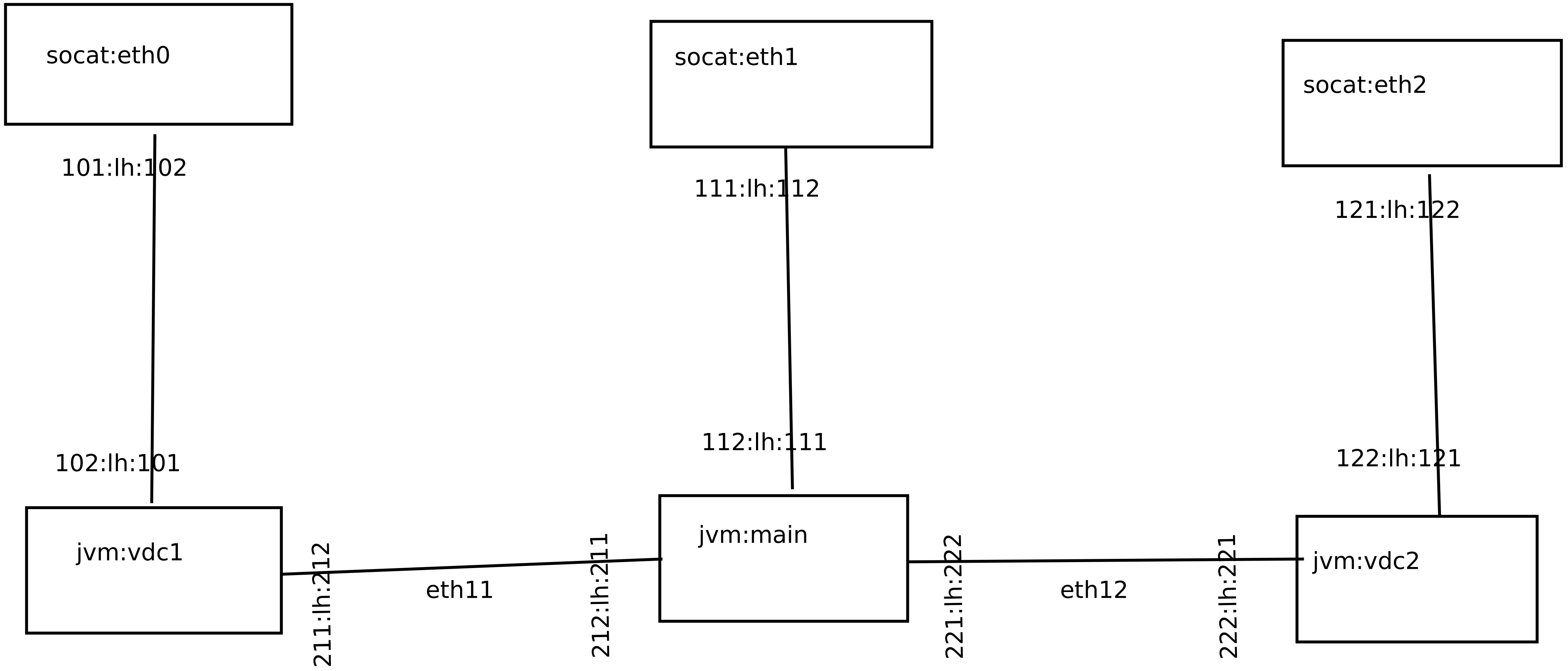
- Single author: http://mc36.nop.hu/cv.html
- Started in 2009 for learning purposes
- Used from 2010 to produce about 30 physical lte routers serving for example as border control's backup infrastructure
- NIIF/Hungarnet use it from 2014 as fullbgp route reflector, bogon and aggregate originator
- About 100 working instances countrywide
- New possibilities are welcomed :)

Code

- Free, open source, automatic changelog since 2012
- Small pieces of functionality in separate classes
- Fully commented, autoformatted source
- Heavily multithreaded design
- Careful refactoring on api interfaces, etc
- Extensive, traffic based self tests run about weekly
- Familiar cli with some differences
- Java7 is the target but compiles cleanly on 6 and 8 with openjdk, ecj, gcj (native)

Architecture

- One unprivileged jvm process (per vdc) that does everything a router does and communicates with the world around it over udp sockets
- Ethernet packets placed back and forth to udp socket with socat linux utility (root required)



Architecture continued

- Vdc; other jvm routers or qemu/kvm images can be started, udp socket passes traffic between them
- A purpose built libpcap based c code which signals ethernet up/downs and outperforms socat about 80%
- A purpose built c code for async hdlc framing (used at the mobile router project)
- Other helpers or table dumps (to asics, openflow, whatever needed) are easily achievable

Features

- Forwarding: ipv4, ipv6, ipx, mpls, layer2 (with irb), atom, eompls, vpls, evpn/pbb
- Routing: ospf, isis, bgp, static, rip, eigrp, babel, bfd, msdp, pim ssm, igmp/mld, and 2 own igp: lsrp, pvrp
- Lsp: p2p, p2mp, mp2mp built by ldp, rsvp or segrou
- Crypto: macsec, ikev1, ikev2, (d)tls1.2, ssh2
- Misc: acl, hqos, nat, vrrp, hsrp, transproxy, 6to4
- Everything is dual stacked and vrf aware from start
- Unlimited encapsulation hierarchy can be applied to a single packet, and it will work... :)

Features continued

- Encapsulation: eth, dot1q, dot1ah, dot1ad, isl, ppp(oe), hdlc, lapb, fr, frppp, atmdxi, atm, isdn
- Tunneling: gre, ipip, l2tp, pptp, gtp, nvgre, vxlan, geneve, mplsip, mplsudp, erspan, etherip, icmptun, pimtun, ipcompress, ipencap, minenc, nos, tmux, l2f, uti, etc
- Servers: telnet, ssh, socks, http(s), ftp(s), tftp, pop3(s), smtp(s), gopher, irc, dcpp, sip, dns, ntp, syslog, lpd, radius, tacacs, etc

Performance

- Few thousand routes and no fancy features usually need same amount of cpu for jvm and socat
- So linux's raw packet handler needs about same number of cpu cycles as the forwarding code
- Jvm7+ uses aes extension of cpus if available so crypto can perform very well
- Jvm optimizes branches in real time so unused code gets optimized out on the fly
- In NIIF/Hungarnet it usually outperforms other's bgp stack
- Throughput depends on cpu power and features but theoretically unlimited (no limiting design patterns)
- 5 year old low end miniitx@100mbps, raspi@10mbps

As1955 uses

- Full v4/v6/msdp feed since 2010
- Primary route reflector since 2014 summer
- Bestpath conqueror since 2015 autumn
- Aggregate&bogon originator since 2015 summer
- Ospf participation for nht to work
- Ldp/traffeng participation for eompls: testbed, sniffings
- Separate vdcs for rr, originator, bogon and testbeds
- (s)afis used: ipv4/ipv6 unicast, multicast, vpn, vpls, evpn, flowspec, vpnflowspec, vpnmulticast, mdt
- Three geographically distributed route reflectors:
- 1st: freerouter, subset, conqueror
- 2nd: freerouter, full tables
- 3rd: other vendor, full tables (no vpls yet)
- Evpn testbed with dante (2 other vendors)
- Other vendor's virtual routers in vdcs form a small lab for interop, prototyping and developing

Questions?

Feel free to find me
see it in live network

RIPE