Linux As A Network OS

Dinesh G Dutt
Shrijeet Mukherjee
Nolan Leake
Pradosh Mohapatra
Networking circa 2008

Flickr photo from [http://www.flickr.com/photos/duncanh1/]
Data Centers have thrown a monkey wrench in the world of networking whether it be how networks are managed, the new technologies they have brought to the fore such as SDN or network virtualization.
Modern data center networks are based around the following ideas:

New breed of applications which are L3-aware and workaround network failures

IP-based networks

Automated management

• Configuration and monitoring
• Rapid spin-up and spin-down of networks
Operating Systems define how you manage the individual boxes and thereby the system
Traditional Router/Switch Oses:

Complex routing/switching features

Structured as a black box:

– No well-defined API

Closed development model

Antediluvian management tool chain

Very slow spin-up and spin-down of networks
Modern data centers usually run GNU/Linux as the server OS:

- Well established and open API
- Sophisticated management tool chain
  - Including scripting
- Vibrant community fueling innovation
- Excellent networking support
Networking support includes the expected and more:
In other words: GNU/Linux is a great fit as the OS for not just servers but even routers and switches in the modern data center
So What? What advantages does this provide?
What Linux as the network OS enables:

Open routing and switching platform
Unified management tool chain
New ways of solving problems
Neatly sidesteps a bunch of problems
A potential to return IP networking to its roots: “rough consensus, working code”
Open Routing/Switching implies a more participatory role for everyone to develop networking, transparency for troubleshooting and understanding.
Open Routing/Switching Platform

Routing Suites

Bridging

Discovery

Monitoring

Quagga

Kernel STP, mstpd

lldpd, open-lldp

Net-snmp, collectd, ganglia
Tools to manage servers such as Chef, Puppet, Ansible etc. can also be used to manage the network.
New ways of solving the problem

Flexlink  →  netplug, ifplugd

HSRP/VRRP  →  Keepalived or a script to configure virtual MAC on bridge
With Linux as the network OS, we neatly sidestep issues that arise from the traditional router OS' being a black box:

Technologies such as netconf APIs for programming network
What's missing?

Hardware acceleration of the networking forwarding path
Switch ports show up as virtual interfaces (swp0, swp1 ...)

- typical tools - ifconfig, route, arp, ip, brctl, ethtool, tcpdump, etc

Kernel FIB/ARP table are synchronized with HW

- userspace can send/receive packets and insert routes normally

Advanced operations use specialized APIs

- expose hardware acceleration beyond stock Linux routing/bridging
Evolving Linux further:

Consistent way to map data structures to hardware

• Netlink's publish functionality not available with netfilter, for eg.

Provide mechanisms to allow commands to fail if backend hardware install fails
## Evolution Of Network OS

<table>
<thead>
<tr>
<th>Monolithic OS</th>
<th>Third Party Real-time OS</th>
<th>Linux-based OS</th>
<th>Linux OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>No real OS, while loop</td>
<td>Embedded OS with process and memory mgmt</td>
<td>Linux as the embedded OS: process and memory mgmt</td>
<td>Linux as Network OS: Native routing and switching</td>
</tr>
<tr>
<td>Proprietary routing And switching stack</td>
<td>Proprietary routing And switching stack</td>
<td>Proprietary routing And switching stack</td>
<td>Cumulus Linux</td>
</tr>
<tr>
<td>Eg: IOS, CatOS</td>
<td>Eg: ION</td>
<td>Eg: NX-OS, EOS</td>
<td></td>
</tr>
</tbody>
</table>
It's taken me all my life to learn what not to play.

- Dizzy Gillespie
Imagine a world where every router ran GNU/Linux
Thank You For Listening!

ddutt@cumulusnetworks.com