

# **Implementation of Business Linux Routers**

Presenter: Joseph Flasch  
jpflasch@gmail.com

# Why Use Linux as a Router ?

- Cost
- Performance
- Reliability
- Open nature of Linux
- It's not IOS
- Multi-function nature of Linux
- Strong networking
- One-box-does-it-all nature of Linux

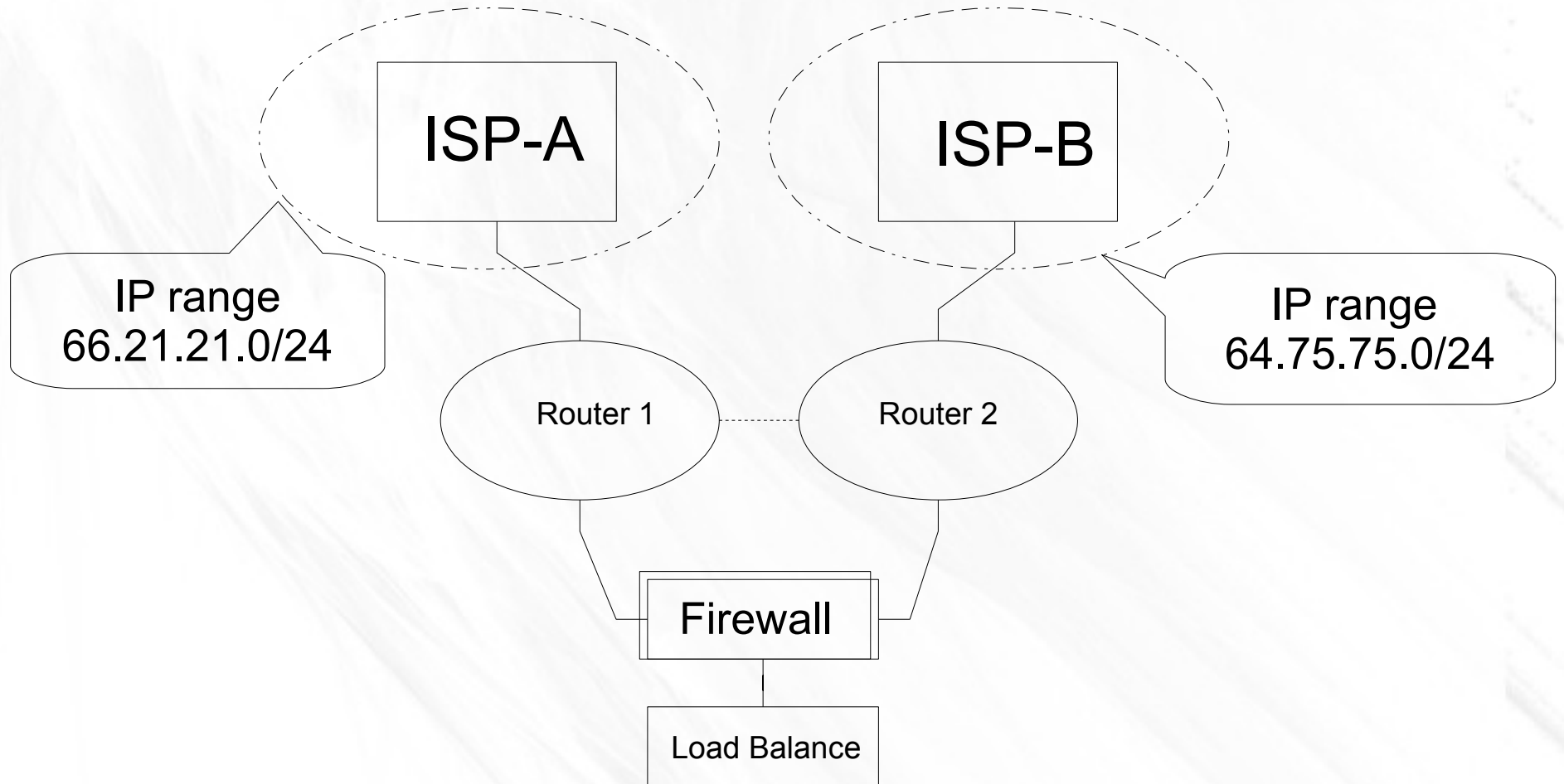
# Tools for a Linux Router

- Zebra / Quagga
  - BGP
  - Metric, weighted, multiple routes
  - OSPF
  - IOS-like
- IP route 2 / Linux kernel / Unix tools
- Iptables / Firewall
- HA Tools, Ultra-Monkey Project / Keep alive
- Tracing tools, network reporting tools

# Zebra / Quagga

- IOS-like Routing Daemons
  - OSPFv2, OSPFv3, RIP v1, v2, RIPng BGP-4
  - Quagga fork of Zebra [www.quagga.net](http://www.quagga.net)
  - TTY type interface language, IOS-like
  - Documentation assumes Cisco experience
  - About 80% like a Cisco router IOS
  - BGP is the work horse of ISP connections
  - Actively supported

# Typical ISP Router connect



# Setting up the Linux Router

- Physical Hardware: Making it work
  - Strong Open Source NIC Drivers
  - Solid Server Hardware, memory
  - Flash-based HDs or raid1 HDs
  - Server BIOS, serial port, TTY access
  - 1U network rack
  - 10 Gig fiber
  - High end switches

# Setting up the Software/Linux

- The Distribution: load it, like it, reload, reload...
  - Can you upgrade? ease of use, philosophy
  - Packages, up to date, feature selection?
  - Red Hat, Debian, Suse, Slackware, Gentoo ...
  - Kernel Building, you should/have to
  - Can you control what gets loaded/started?
  - Setting up network daemons, Quagga
  - SSH access, key based , IP based
  - TTY console, TTY Zebra, BGP access

# Kernel Building 101

- Set up Kernel CPU / NIC / ACPI / Network

```
<*) Packet socket
[*] Packet socket: mmaped IO
<*) Unix domain sockets
< > PF_KEY sockets
[*] TCP/IP networking
[ ] IP: multicasting
[*] IP: advanced router
    Choose IP: FIB lookup algorithm (choose FIB_HASH if unsure) (FIB_HASH) --->
[*] IP: policy routing
[*] IP: equal cost multipath
[*] IP: verbose route monitoring
[*] IP: kernel level autoconfiguration
[ ] IP: DHCP support
[ ] IP: BOOTP support
[ ] IP: RARP support
<M> IP: tunneling
<M> IP: GRE tunnels over IP
[ ] IP: ARP daemon support (EXPERIMENTAL)
[ ] IP: TCP syncookie support (disabled per default)
< > IP: AH transformation
< > IP: ESP transformation
< > IP: IPComp transformation
< > IP: IPsec transport mode
< > IP: IPsec tunnel mode
u(+)
```

Iproute2 needs this

Ipsec/Tun/Gre



```
v(+)  
< > IP: IPsec BEET mode  
< > Large Receive Offload (ipv4/tcp)  
< > INET: socket monitoring interface  
[ ] TCP: advanced congestion control --->  
[ ] TCP: MD5 Signature Option support (RFC2385) (EXPERIMENTAL)  
< > The IPv6 protocol --->  
[ ] Security Marking  
[*] Network packet filtering framework (Netfilter) --->  
< > The DCCP Protocol (EXPERIMENTAL) --->  
< > The SCTP Protocol (EXPERIMENTAL) --->  
< > The TIPC Protocol (EXPERIMENTAL) --->  
<M> Asynchronous Transfer Mode (ATM)  
<M> Classical IP over ATM  
[ ] Do NOT send ICMP if no neighbour  
< > LAN Emulation (LANE) support  
< > RFC1483/2684 Bridged protocols  
<M> 802.1d Ethernet Bridging  
[ ] Distributed Switch Architecture support --->  
<M> 802.1Q VLAN Support  
[ ] GVRP (GARP VLAN Registration Protocol) support  
< > DECnet Support  
<M> ANSI/IEEE 802.2 LLC type 2 Support  
< > The IPX protocol  
< > Appletalk protocol support  
v(+)
```

Bridge support

Vlan Support

**<Select>** < Exit > < Help >

## NETWORK packet filtering framework (Netfilter)

Arrow keys navigate the menu, <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [\*] built-in [ ] excluded <M> module < > module capable

```
-- Network packet filtering framework (Netfilter)
[ ] Network packet filtering debugging
[*] Advanced netfilter configuration
[*] Bridged IP/ARP packets filtering
Core Netfilter Configuration --->
< > IP virtual server support --->
IP: Netfilter Configuration --->
<M> Ethernet Bridge tables (eatables) support --->
```



Iptables

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [\*] built-in [ ] excluded <M> module < > module capable

```
<M> Netfilter NFQUEUE over NFNETLINK interface
<M> Netfilter LOG over NFNETLINK interface
<M> Netfilter connection tracking support
  *- Connection tracking flow accounting
  *- Connection mark tracking support
  [*] Connection tracking events
  < > DCCP protocol connection tracking support (EXPERIMENTAL)
  <M> SCTP protocol connection tracking support (EXPERIMENTAL)
  <M> UDP-Lite protocol connection tracking support
  <M> Amanda backup protocol support
  <M> FTP protocol support
  <M> H.323 protocol support
  <M> IRC protocol support
  <M> NetBIOS name service protocol support
  <M> PPTP protocol support
  <M> SANE protocol support (EXPERIMENTAL)
  <M> SIP protocol support
  <M> TFTP protocol support
  <M> Connection tracking netlink interface
  < > Transparent proxying support (EXPERIMENTAL)
{M} Netfilter Xtables support (required for ip_tables)
  <M> "CLASSIFY" target support
  < > "CONNMARK" target support
  < > "DSCP" and "TOS" target support
  <M> "MARK" target support
v(+)
```

State full FW  
Protocols

<Select> < Exit > < Help >

## IP: Netfilter Configuration

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Press <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. L [ ] excluded <M> module < > module capable

```
<M> IPv4 connection tracking support (required for NAT)
[*]  proc/sysctl compatibility with old connection tracking
<M>  IP Userspace queueing via NETLINK (OBSOLETE)
<M>  IP tables support (required for filtering/masq/NAT)
<M>  "addrtype" address type match support
<M>  "ah" match support
<M>  "ecn" match support
<M>  "ttl" match support
<M>  Packet filtering
<M>  REJECT target support
<M>  LOG target support
<M>  ULOG target support
<M>  Full NAT
<M>  MASQUERADE target support
<M>  NETMAP target support
<M>  REDIRECT target support
<M>  Basic SNMP-ALG support
<M>  Packet mangling
<M>  CLUSTERIP target support (EXPERIMENTAL)
<M>  ECN target support
<M>  TTL target support
<M>  raw table support (required for NOTRACK/TRACE)
<M>  ARP tables support
<M>  ARP packet filtering
<M>  ARP payload mangling
```



NAT: Dnat Snat

## Ethernet Bridge tables (ebtables) support

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are  
<N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for  
[ ] excluded <M> module < > module capable

### --- Ethernet Bridge tables (ebtables) support

```
<M> ebt: broute table support
<M> ebt: filter table support
<M> ebt: nat table support
<M> ebt: 802.3 filter support
<M> ebt: among filter support
<M> ebt: ARP filter support
<M> ebt: IP filter support
<M> ebt: limit match support
<M> ebt: mark filter support
<M> ebt: packet type filter support
<M> ebt: STP filter support
<M> ebt: 802.1Q VLAN filter support
<M> ebt: arp reply target support
<M> ebt: dnat target support
<M> ebt: mark target support
<M> ebt: redirect target support
<M> ebt: snat target support
<M> ebt: log support
<M> ebt: ulog support (OBSOLETE)
< > ebt: nflog support
```

# Setup of the Network parts

```
! Zebra configuration saved from vty
```

```
! 2008/06/05 05:21:02
```

```
!
```

```
hostname Router
```

```
password verybigpw
```

```
enable password verbigpw
```

```
log stdout
```

```
log syslog
```

```
!
```

```
interface eth0
```

```
shutdown
```

```
interface lo
```

```
!
```

```
!  
interface vlan100  
  description My ISP info phone # ticket instructions etc  
  ip address 109.16.19.129/29  
  ipv6 nd suppress-ra  
!  
interface vlan200  
  ip address 10.129.28.50/24  
  ipv6 nd suppress-ra  
  
!  
access-list 10 permit 192.168.1.0 0.0.0.255  
!  
ip forwarding  
ip route 0.0.0.0/0 10.199.128.221 200  
ip route 0.0.0.0/0 10.199.128.2 205  
ip route 65.44.42.0 255.255.255.0 10.129.28.1  
ip route 68.17.188.0 255.255.255.0 10.129.28.1  
!  
line vty  
!
```

# BGP Config

```
router bgp 77688
```

ASA #

```
bgp router-id 217.201.249.2
```

```
network 217.201.249.0/25
```

```
network 64.87.141.0/24
```

```
network 67.128.177.0/24
```

```
neighbor ibgp-eB peer-group
```

```
neighbor ibgp-eB remote-as 77688
```

```
neighbor ibgp-eB next-hop-self
```

```
neighbor ibgp-eB default-originate
```

```
neighbor ibgp-eB soft-reconfiguration inbound
```

```
neighbor ibgp-eB route-map INT_WO_PRE out
```

```
neighbor ibgp-eB filter-list 6 out
```

Floating ip ranges

Internal BGP  
group def.



# BGP Internal

```
neighbor ibgp-eb filter-list 6 out
neighbor 10.199.128.251 peer-group ibgp-eb
description 221 is the secondary site1 router
neighbor 10.252.1.221 peer-group ibgp-eb
neighbor 10.252.1.221 weight 11
description 222 is the primary verizon router
neighbor 10.252.1.222 peer-group ibgp-eb
neighbor 10.252.1.222 weight 12
description 242 is the secondary site2 router
neighbor 10.252.1.242 peer-group ibgp-eb
neighbor 10.252.1.242 weight 9
```

Neighbor statements:  
Note the use of group  
ibgp-eb and weight

# BGP external

```
neighbor ebgp-eb peer-group
neighbor ebgp-eb remote-as 6461
neighbor ebgp-eb soft-reconfiguration inbound
neighbor ebgp-eb route-map AB_net_IN in
neighbor ebgp-eb route-map AB_net_Out_PRE out
neighbor ebgp-eb weight 300
neighbor 212.66.199.226 peer-group ebgp-eb
neighbor 212.66.199.227 peer-group ebgp-eb
```

# BGP Filters

```
access-list 15 permit 216.200.249.0 0.0.0.128
access-list 25 permit 66.117.177.0 0.0.0.255
access-list 25 permit 63.86.141.0 0.0.0.255
access-list 35 permit 216.200.249.0 0.0.0.128
```

Like Cisco Access List builds IP filters for allowing IP ranges

```
!
ip as-path access-list 6 permit ^$
ip as-path access-list 8 permit ^$
ip as-path access-list 8 permit .*
```

Regx expressions

```
!
route-map AB_net_Out_PRE permit 20
  match ip address 25
  set as-path prepend 77688 77688 77688
```

The longer the path, the more the path will not be used

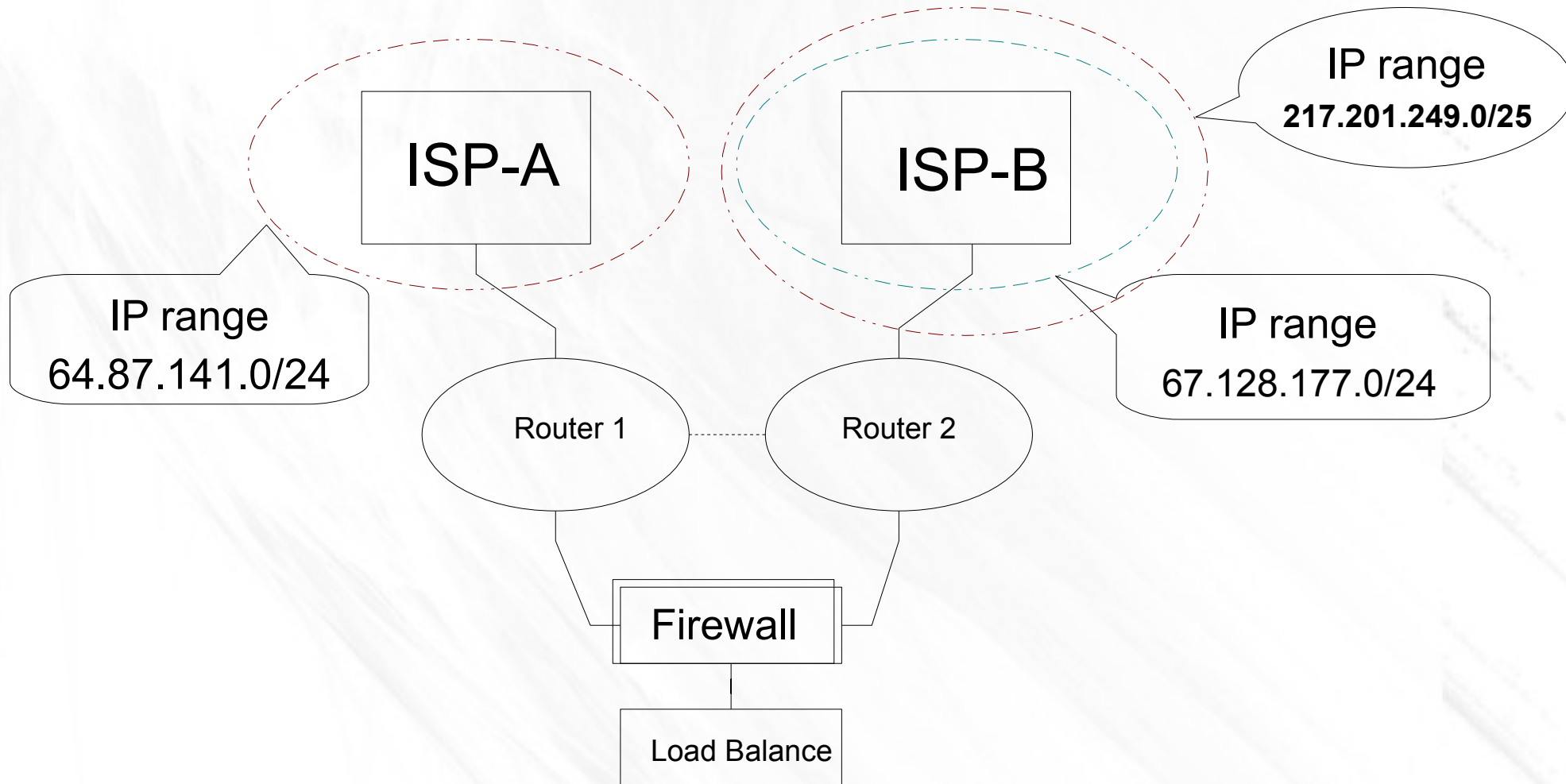
```
!
route-map AB_net_Out_PRE permit 30
  match ip address 15
```

Host this IP range

```
!
route-map INT_WO_PRE permit 20
  match ip address 35
```

```
!
route-map AB_net_IN permit 10
  match as-path 8
```

# Typical ISP Router connect



# BGP Summary

- Used to get the default route from ISP
- Used to manage active ISP IP Ranges
- Used to manage groups of routers
- Problems with BGP
  - Old, well-supported, but not as nice as OSPF
  - BGP ISO support language is hard to understand

# Linux Firewall

- Input, Output and Forward queues
- Nat, Dnat, Snat and MASQUERAD
- Mangle, a packet
- Load Balance
- Map IP to IP ranges
- Randomize to a dest
- And more ... Very active development in the Kernel

# Linux LB (IP virtual server)

- IP virtual server, in the Linux kernel since 2.4
  - Many Load Balance types
    - round-robin scheduling
    - weighted round-robin scheduling
    - least-connection scheduling
    - weighted least-connection scheduling
    - locality-based least-connection scheduling
    - locality-based least-connection with replication scheduling
    - destination hashing scheduling
    - source hashing scheduling
    - shortest expected delay scheduling
    - never queue scheduling

# Using IP Virtual Server

- Ipvssadm – base package to control IP VS
- HA Heart Beat or Keepalive to control IP VS
- HA uses Ld director perl script to control VIP and target hosts, and test if active
- Ld director will test many types of services, lots of flexible options for testing



# Conclusions, Observations

- The Linux platform opens networking up to many normal Unix administration employees, whereas Cisco networking is very specialized and can take years to learn. Many small businesses can't handle this.
- The equipment cost savings can be huge at high bandwidth rates, and taking ownership of your network has many other advantages.
- Upgrade of software is easy and painless.
- Combining routers with FW/LB is possible .

# More info on Topics

- Zebra/Quagga - [quagga.net](http://quagga.net), [zebra.org](http://zebra.org)
- BGP – O'Reilly BGP
- Iptables/Netfilter - [netfilter.org](http://netfilter.org)
- HA Project - [linux-ha.org](http://linux-ha.org)
- IP route2 - [linuxfoundation.org/en/Net:Iproute2](http://linuxfoundation.org/en/Net:Iproute2)
- Keep alive - [www.keepalived.org](http://www.keepalived.org)