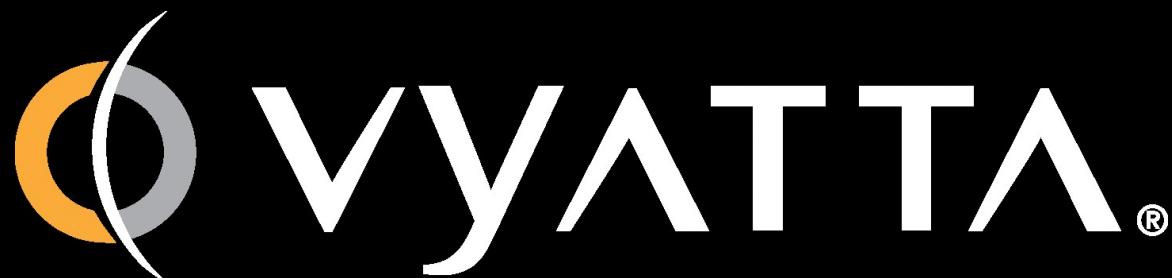


Herd^{ing} networking cats: Integrating Linux routing with FusionCLI™

Stephen Hemminger
shemminger@vyatta.com



<http://vyatta.org>

- Free download
 - livecd
- Update 2x year
- Community forums

<http://vyatta.com>

- Subscription
- Update 4x year
- Software or Hardware
- Phone and Email Support

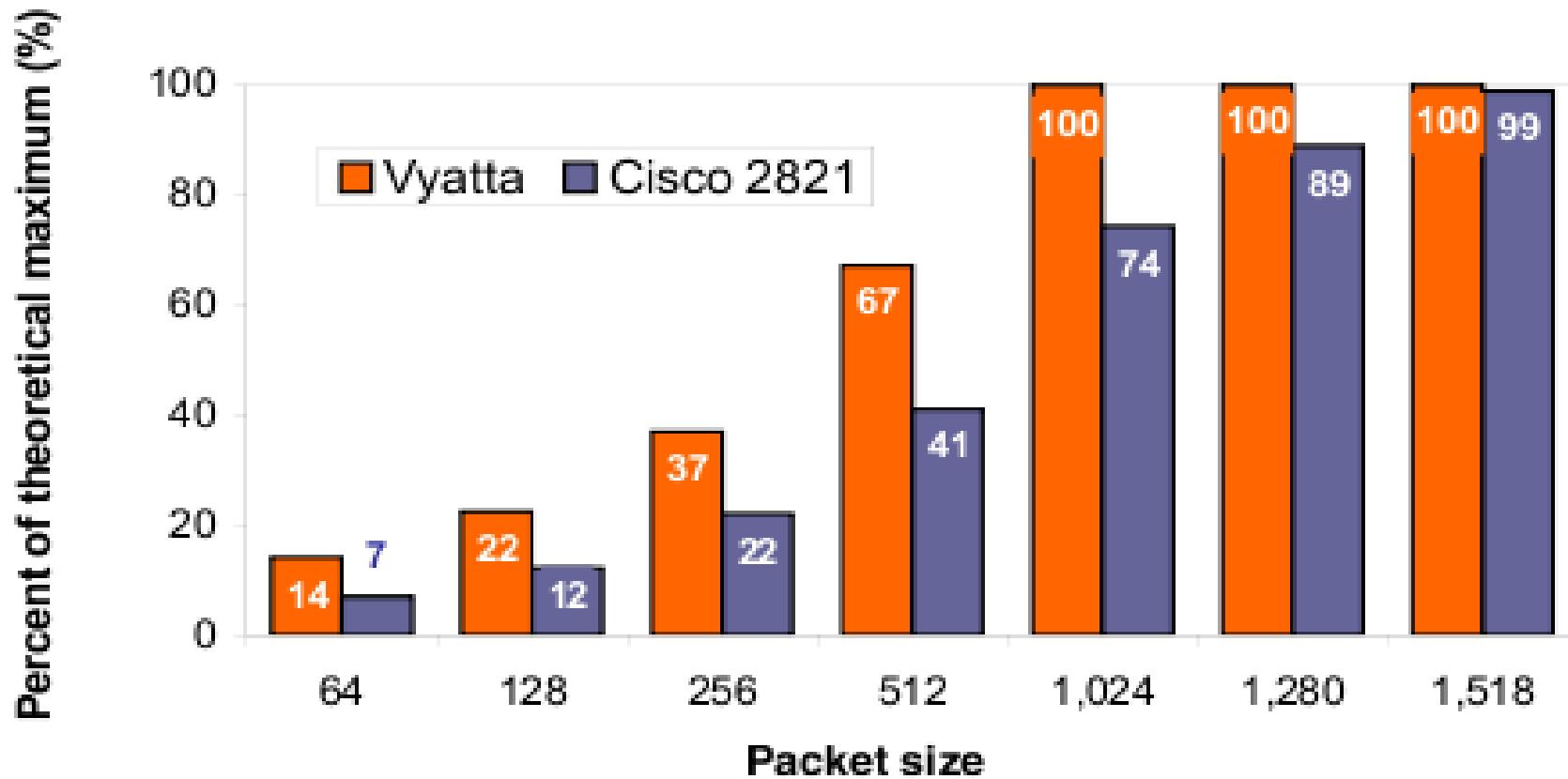
Same source and features

Our customer?



Zero-Loss (≤ 0.001) Bidirectional Routing Throughput Across Two Gigabit Ethernet Ports (100 Flows)

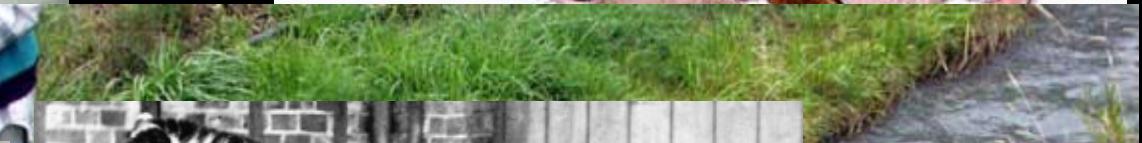
as Reported by Spirent SmartFlow 5.5



Source: The Tolly Group, February 2007

Figure 1

Linux cat



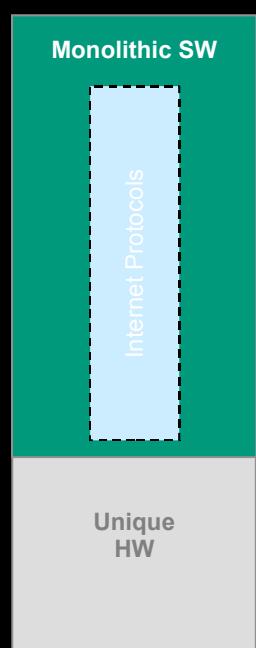
Worth 1000.com



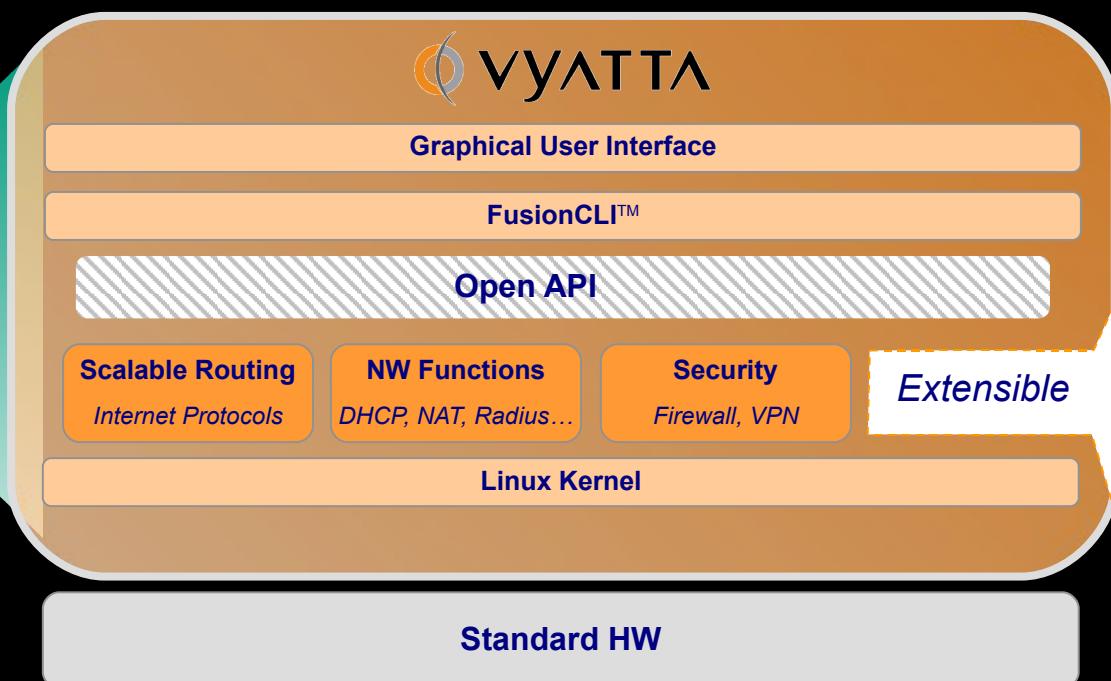
- Ethernet → ifconfig, ip link
- WAN → wanpipe
- Routing → route, vtysh, ip route
- Bridging → brctl
- VLAN → vconfig, ip link
- Bonding → ifenslave
- User mgmt → adduser
- VPN
- QoS → tc

- Router look & feel
 - Command completion
 - Roles: Administrator, operator
 - Configure mode
- Extensible
 - Text based
 - Language neutral
- Integrated with operating system

Proprietary
System



Open
System



Network Apps

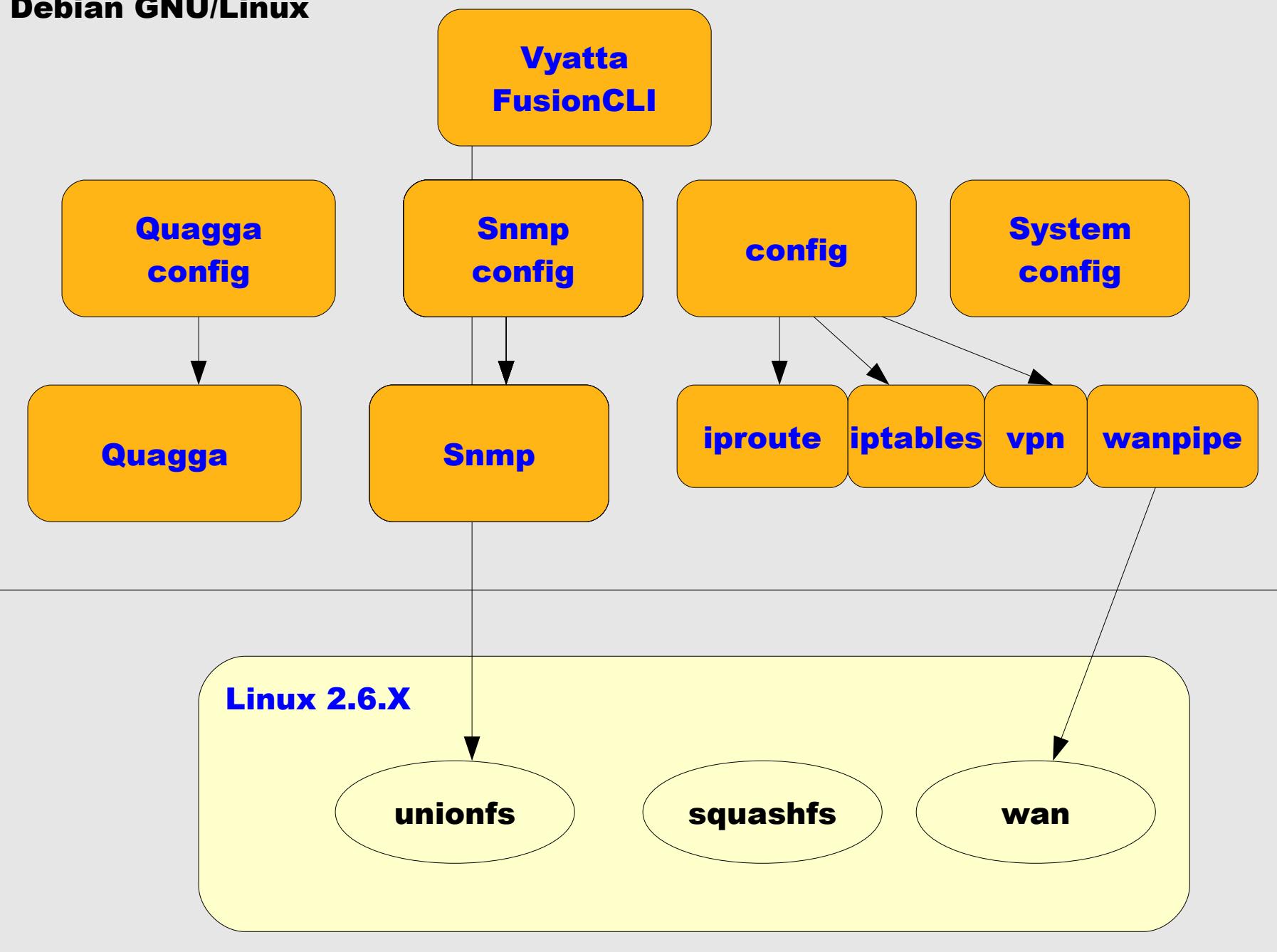
WAN Optimization

Load Balancing

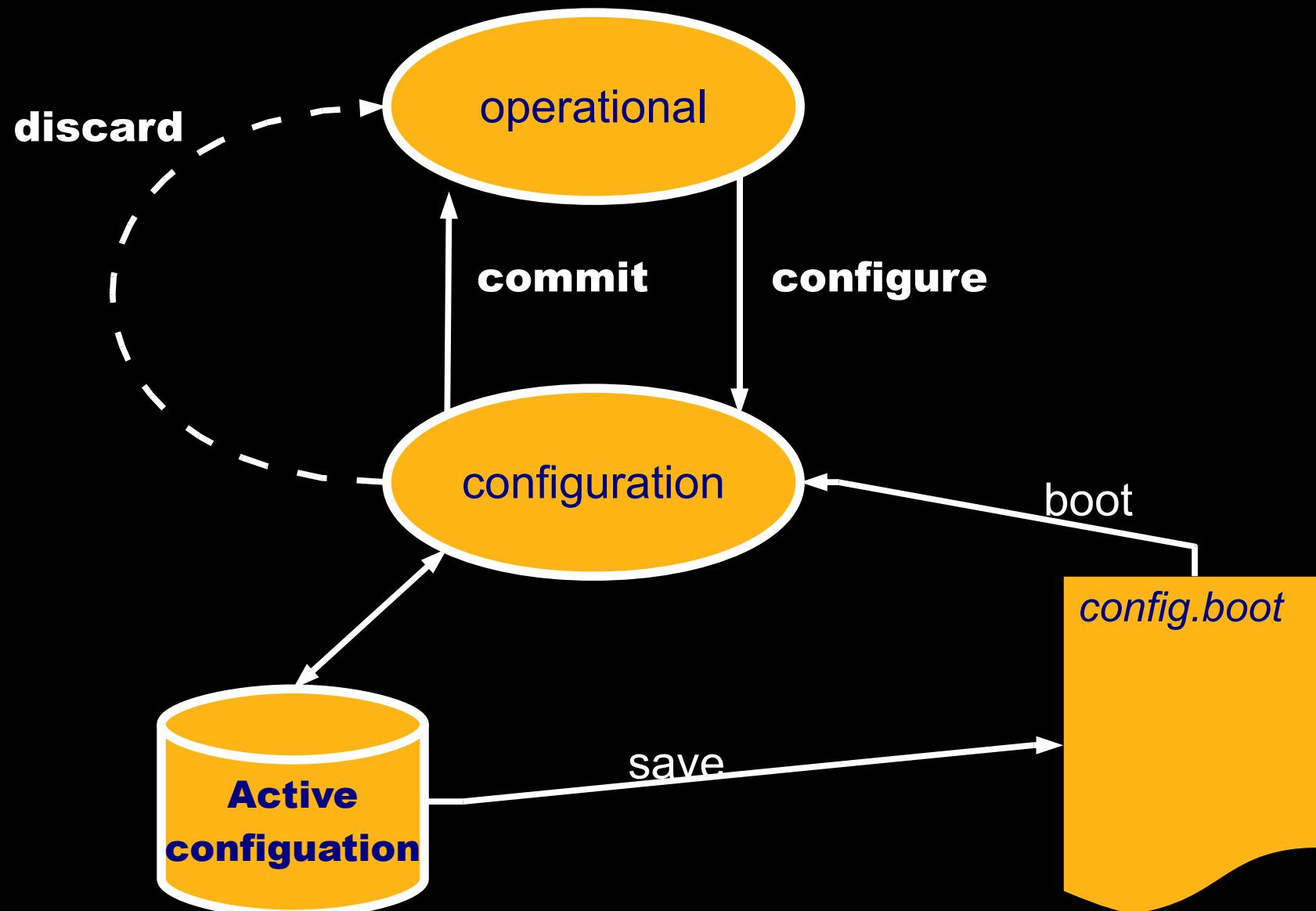
PBX Anti-X

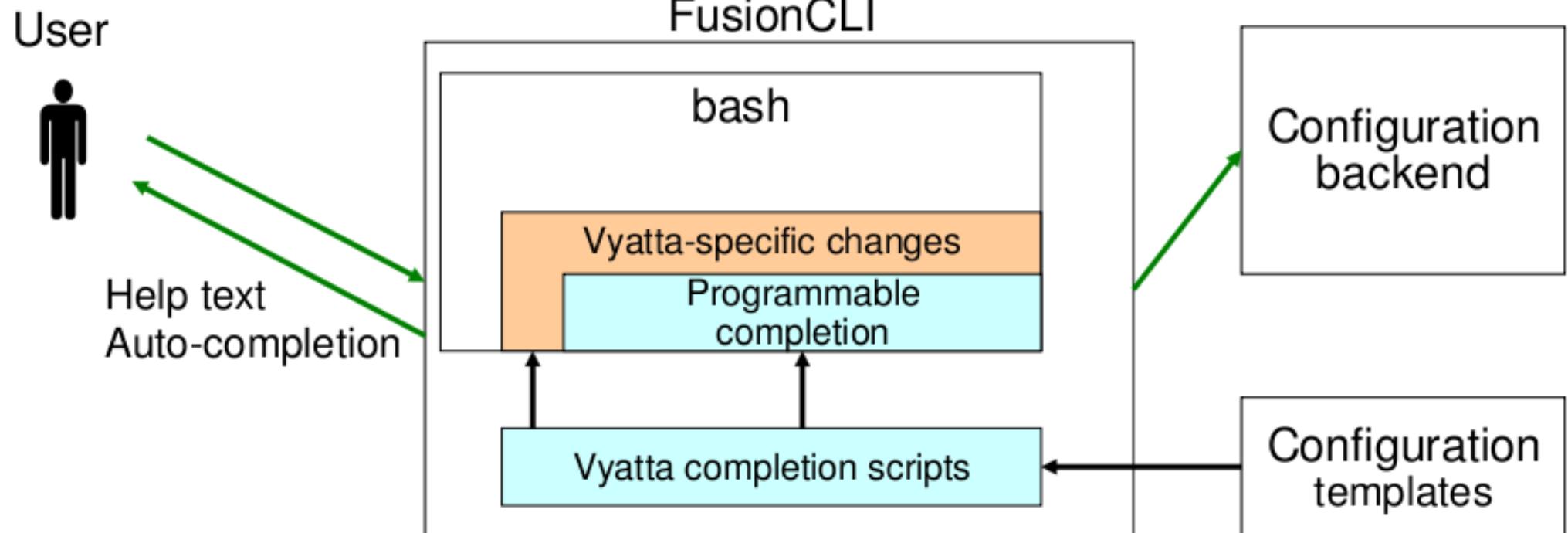
IDS

Massive open-source
ecosystem...

Debian GNU/Linux

CLI configuration modes





Demo 1: Basic interface



```
$ show interfaces
Interface      IP Address          State      Link      Description
eth0           192.168.111.132/24 up        up
eth1           -                  up        up
lo             127.0.0.1/8       up        up
lo             ::1/128          up        up

$ show interfaces et<tab>hernet <tab>
detail    eth0      eth1

$ show interfaces ethernet eth0
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UNKNOWN qlen 1000
link/ether 00:0c:29:f6:20:9e brd ff:ff:ff:ff:ff:ff
inet 192.168.111.132/24 brd 192.168.111.255 scope global eth0
inet6 fe80::20c:29ff:fe:f6:209e/64 scope link
    valid_lft forever preferred_lft forever
...
...
```

Template hierarchy



```
/opt/vyatta/share/vyatta-op/templates/show/interfaces/ethernet
|-- detail
|   `-- node.def
|-- node.def
`-- node.tag
  |-- brief
  |   `-- node.def
  |-- capture
  |   |-- node.def
  |   |-- not
  |       |-- node.def
  |       `-- port
  |           |-- node.def
  |           `-- node.tag
  |               `-- node.def
  |-- port
  |   |-- node.def
  |   `-- node.tag
  |       `-- node.def
  |-- identify
  |   `-- node.def
  |-- node.def
  |-- physical
  |   `-- node.def
```

show/interfaces/ethernet/node.tag/node.def

```
help: Show specified ethernet interface information  
allowed: for dev in /sys/class/net/*;  
        do if [[ -L $dev/device ]]  
            then if [[ $(cat $dev/type) -eq 1 ]]  
                then echo -n ${dev##*/} "  
            fi  
        fi  
done  
run: vyatta-show-interfaces.pl --intf="$4"
```

Demo 2: Configuration



```
$ configure
[edit]
# set interfaces ethernet eth0 description 'Vmware NAT'
[edit]
# show interfaces ethernet eth0
  address dhcp
+description "Vmware NAT"
  hw-id 00:0c:29:f6:20:9e
[edit]

# mount
...
unionfs on /opt/vyatta/config/tmp/new_config_5035 type unionfs
(rw,dirs=/tmp/changes_only_5035=rw:/opt/vyatta/config/active=ro)
[edit]
vyatta@vc3-1-3# commit
```

Configuration templates

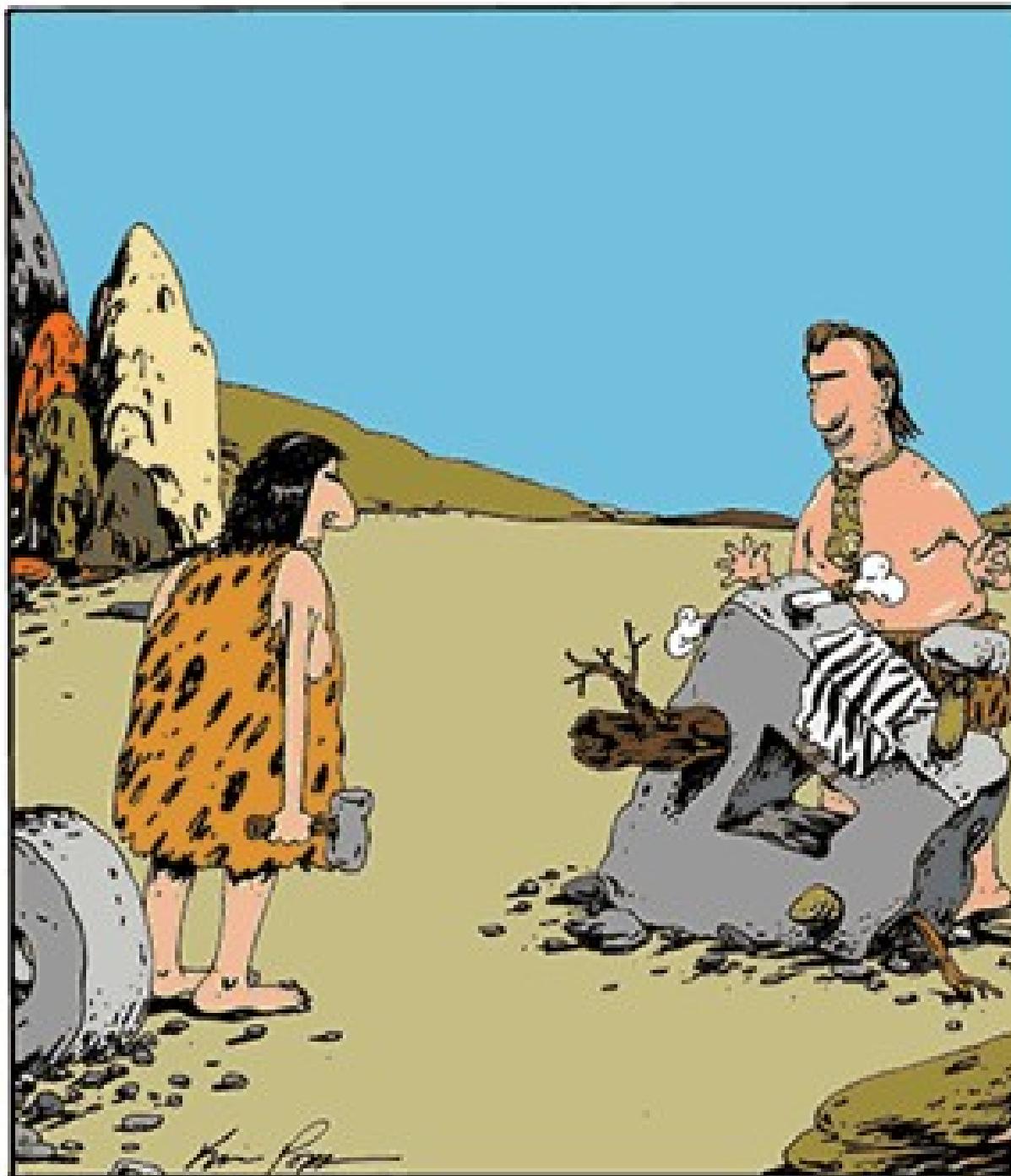


```
multi:  
type: txt  
help: Set an IP address for this interface  
syntax:expression: exec "/opt/vyatta/sbin/vyatta-interfaces.pl \  
--valid-addr $VAR(@) --dev $VAR(../@)" \  
; "Invalid IP address/prefix [$VAR(@)] for interface $VAR(../@)"  
  
update: /opt/vyatta/sbin/vyatta-interfaces.pl \  
--eth-addr-update $VAR(@) --dev $VAR(../@)  
  
delete: /opt/vyatta/sbin/vyatta-interfaces.pl \  
--eth-addr-delete $VAR(@) --dev $VAR(../@)  
  
allowed: echo "dhcp <>"  
comp_help:Possible completions:  
<x.x.x.x/x>      Set the IP address and prefix length  
<h:h:h:h:h:h:h/x>  Set the IPv6 address and prefix length  
dhcp              Set the IP address and prefix length via DHCP
```

Configuration save restore



```
interfaces {  
    ethernet eth0 {  
        address dhcp  
        duplex auto  
        hw-id 00:0c:29:f6:20:9e  
        speed auto  
    }  
    loopback lo {  
    }  
}  
service {  
    ssh {  
        port 22  
        protocol-version v2  
    }  
}
```



If Cisco had invented the wheel.

- Real time services
 - VOIP
- Network control plane
 - BGP, OSPF, STP
- Fairness
- Throttle batch services
 - P2P, backup,

- Organized by policy types
 - Fair queue => sfq
 - Traffic shaper => htb
 - Drop tail => fifo
 - Rate limiter => tbf
 - Traffic limiter => ingress

...

```
# set qos-policy fair-queue fq
[edit]
# set interfaces ethernet eth0 qos-policy out fq
[edit]
# commit
[edit]
# run show queueing
Output queues:
Interface  Qos-Policy          Sent      Dropped      Overlimit
eth0        fair-queue          4578       0            0
eth1        default             468        0            0
```

set/qos-policy/fair-queue/node.def

tag:

type: txt

help: Set fair queueing policy

syntax:expression: pattern \$VAR(@) "^[[:alnum:]][-_[:alnum:]]*\$"
; "only alpha-numeric policy name allowed"

update: /opt/vyatta/sbin/vyatta-qos.pl --create-policy "\$VAR(.)"
"\$VAR(@)"

delete: /opt/vyatta/sbin/vyatta-qos.pl --delete-policy "\$VAR(@)"

```
set/interfaces/ethernet/node.tag/qos-policy/out/node.def
```

```
type: txt
help: Set outbound QOS policy for specified ethernet interface
allowed: /opt/vyatta/sbin/vyatta-qos.pl --list-policy
update: /opt/vyatta/sbin/vyatta-qos.pl \
        --update-interface $VAR(..../@) $VAR(.) $VAR(@)
delete: /opt/vyatta/sbin/vyatta-qos.pl \
        --delete-interface $VAR(..../@) $VAR(.)
```

```
sub update_interface {
    my ($interface, $direction, $name) = @_;
    my $config = new VyattaConfig;

    ( $direction eq "out" ) or die "Only out direction supported";

    $config->setLevel('qos-policy');
    foreach my $type ( $config->listNodes() ) {
        if ( $config->exists("$type $name") ) {
            my $shaper = make_policy($config, $type, $name);
            delete_interface($interface, $direction);
            open my $out, "|-" or exec qw:sudo /sbin/tc -batch -:;
            $shaper->commands($out, $interface);
            if (! close $out) {
                delete_interface($interface, $direction);
            }
        }
        die "Unknown qos-policy $name\n";
    }
}
```

```
# edit qos-policy traffic-shaper lartc  
[edit qos-policy traffic-shaper lartc]  
  
# set class 2 bandwidth 100%  
  
[edit qos-policy traffic-shaper lartc]  
  
# set class 2 match www ip destination port 80  
  
[edit qos-policy traffic-shaper lartc]  
  
# set class 3 bandwidth 3mbit  
  
[edit qos-policy traffic-shaper lartc]  
  
# set class 3 ceiling 5mbit  
  
[edit qos-policy traffic-shaper lartc]  
  
# set class 3 match smtp ip destination port 25  
  
[edit qos-policy traffic-shaper lartc]  
  
# exit
```

Traffic-shaper continue



```
[edit]
```

```
# commit
```

```
qos-policy traffic-shaper lartc configuration not complete: missing  
default class
```

```
Commit failed
```

```
# set qos-policy traffic-shaper lartc default bandwidth 1
```

```
[edit]
```

```
# commit
```

```
[edit]
```

```
# set interfaces ethernet eth0 qos-policy out
```

```
# run show queueing ethernet eth0
```

```
eth0 Output queue:
```

Class	Qos-Policy	Sent	Dropped	Overlimit
1:	traffic-shaper	11438	0	0
4	fair-queue	11438	0	0
2	fair-queue	0	0	0
3	fair-queue	0	0	0

```
[edit]
```

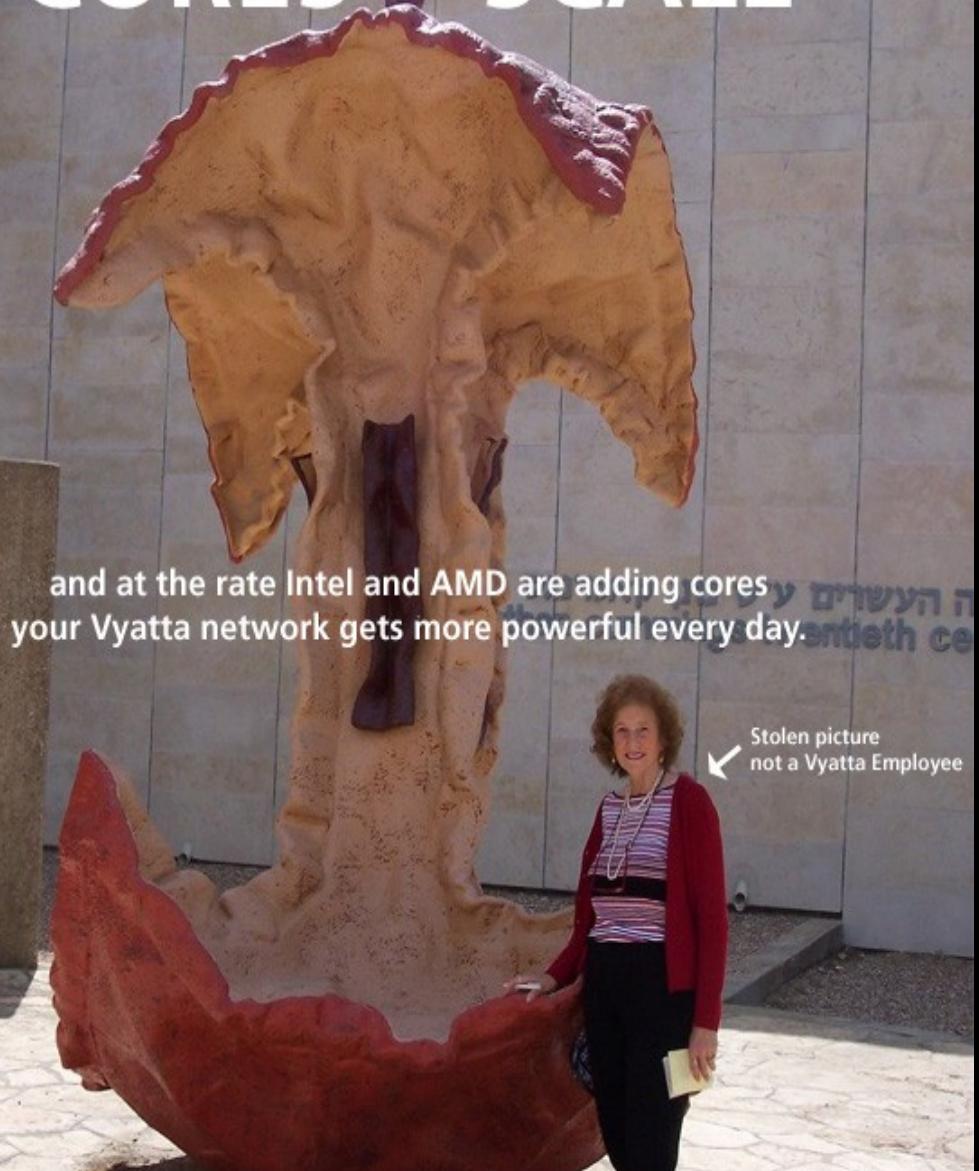
- Vyatta package changes
 - Every distribution is a fork
 - All changes are fed to upstream
- Vyatta ↔ Linux management
 - Vyatta config ignores other changes
 - CLI is loosely coupled
- Developer documentation
 - Linux Kongress paper
 - Watch this space

- Users want support for orphaned projects
 - Multicast routing
 - MPLS
 - Rapid Spanning Tree Protocol
 - IPV6



For Vyatta users

CORES = SCALE



Future

- Richer features
 - QoS+, bonding, ...
- GUI
- Multi-queue Performance