Crossbow Virtual Wire: Network In a Box

Sunay Tripathi, Nicolas Droux,
Kais Belgaied, Shrikrishna Khare

November 5th, 2009
USENIX LISA 09, Baltimore, MD

Nicolas Droux, Senior Staff Engineer
Solaris Kernel Networking, Sun Microsystems Inc.
nicolas.droux@sun.com

Key Issues in Network Virtualization

- Fair or Policy based resource sharing in virtualized environments
  - Bandwidth
  - NIC Hardware resources including Rx/Tx descriptors
  - Processing CPUs
- Overheads due to Virtualization
  - Latency, Throughput
- Management
  - Isolation between distributed applications
  - Network fabric configuration
- Security
  - New threats to L2 network
- Where to solve the problem?
  - Switches
  - L3/L4 devices
  - Hosts

Crossbow: Solaris Networking Stack

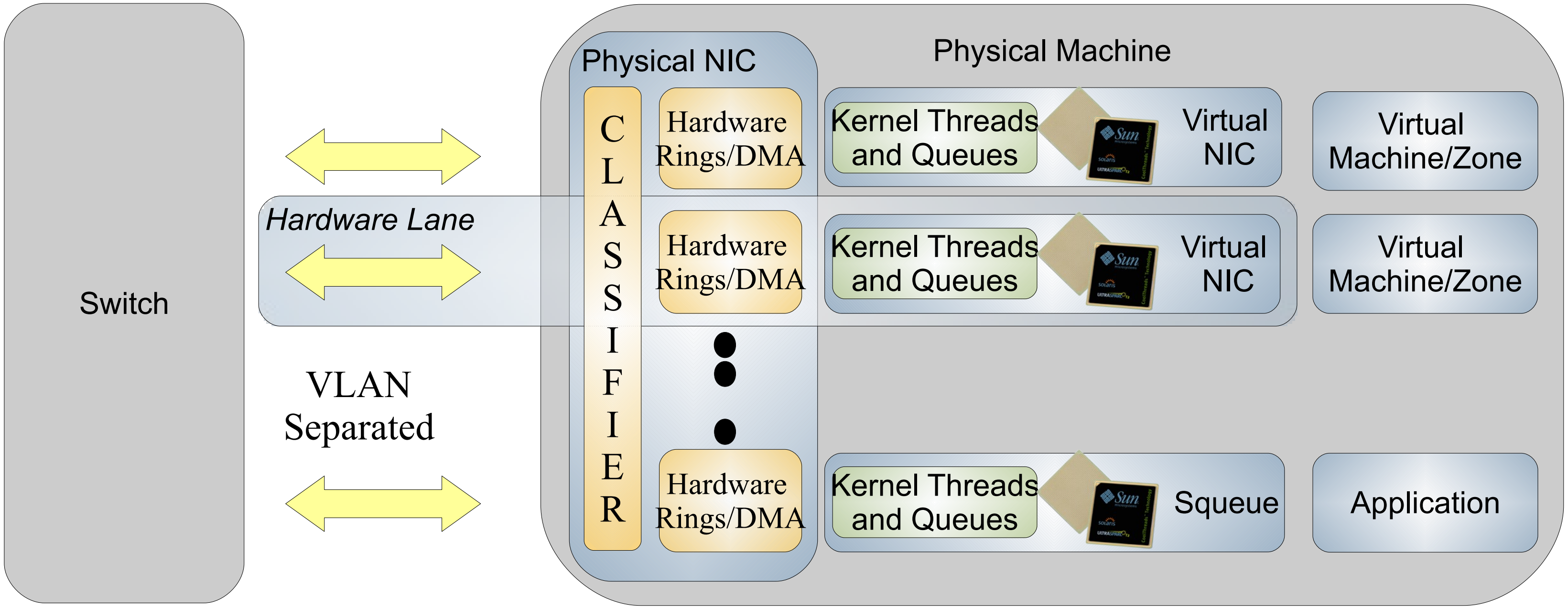
- 8 years of development work to achieve
  - > Scalability across multi-core CPUs and multi-10gigE bandwidth
  - > Virtualization, QoS, High-availability designed in
  - > Exploit advanced NIC features

- Key Enabler for
  - > Server and Network Consolidation
  - > Open Networking
  - > Cloud computing

Crossbow “Hardware Lanes”

Ground-Up Design for multi-core and multi-10GigE

- Linear Scalability using '**Hardware Lanes**' with dedicated resources
- Network Virtualization and QoS designed in the stack
- More Efficiency due to '**Dynamic Polling and Packet Chaining**'



Hardware Lanes and Dynamic Polling

- Partition the NIC Hardware (Rx/Tx rings, DMA), kernel queues/threads, and CPU to allow creation of “Hardware Lane” which can be assigned to VNICs & Flows
- Use Dynamic Polling on Rx/Tx rings to schedule rate of packet arrival and transmission on a per lane basis
- Effect of dynamic polling

Mpstat (older driver)

```
intr   ithr  csw   icsw  migr smtx  srw  syscl  usr sys wt idl
10818  8607  4558  1547  161  1797  289  19112  17  69  0  12
```

Mpstat (GLDv3 based driver)

```
intr   ithr  csw   icsw  migr smtx  srw  syscl  usr sys wt idl
2823   1489  875   151   93   261   1    19825  15  57  0  27
```

~75% Fewer Interrupts

~85% Fewer Ctx Switches

~85% Fewer Mutexes

~15% More CPU Free

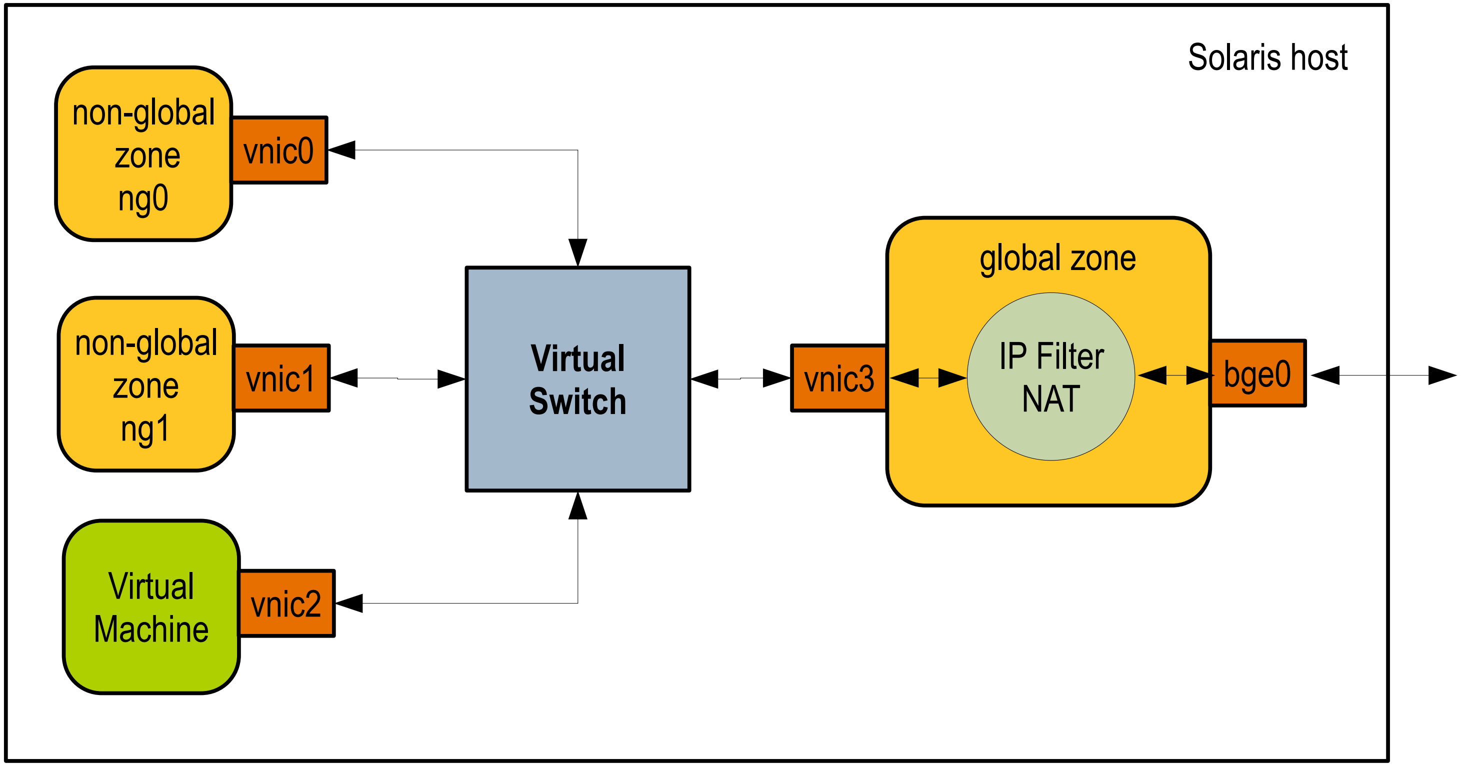
Crossbow Virtual NICs (VNICs)

- Pseudo MAC instances
  - Can be managed as if they were physical NICs
  - Per VNICs stats, reuse existing management tools
  - Link speed derived from configured bandwidth limit
  - High-Availability by creating VNICs on link aggregations or combining VNICs in IPMP groups
- Dedicated per-VNIC hardware and kernel resources
- Data path pass-through, no bump in the stack
- Standards based End-to-End Network Virtualization
  - VLAN tags and Priority Flow Control (PFC) assigned to VNIC extend Hardware Lanes to Switch

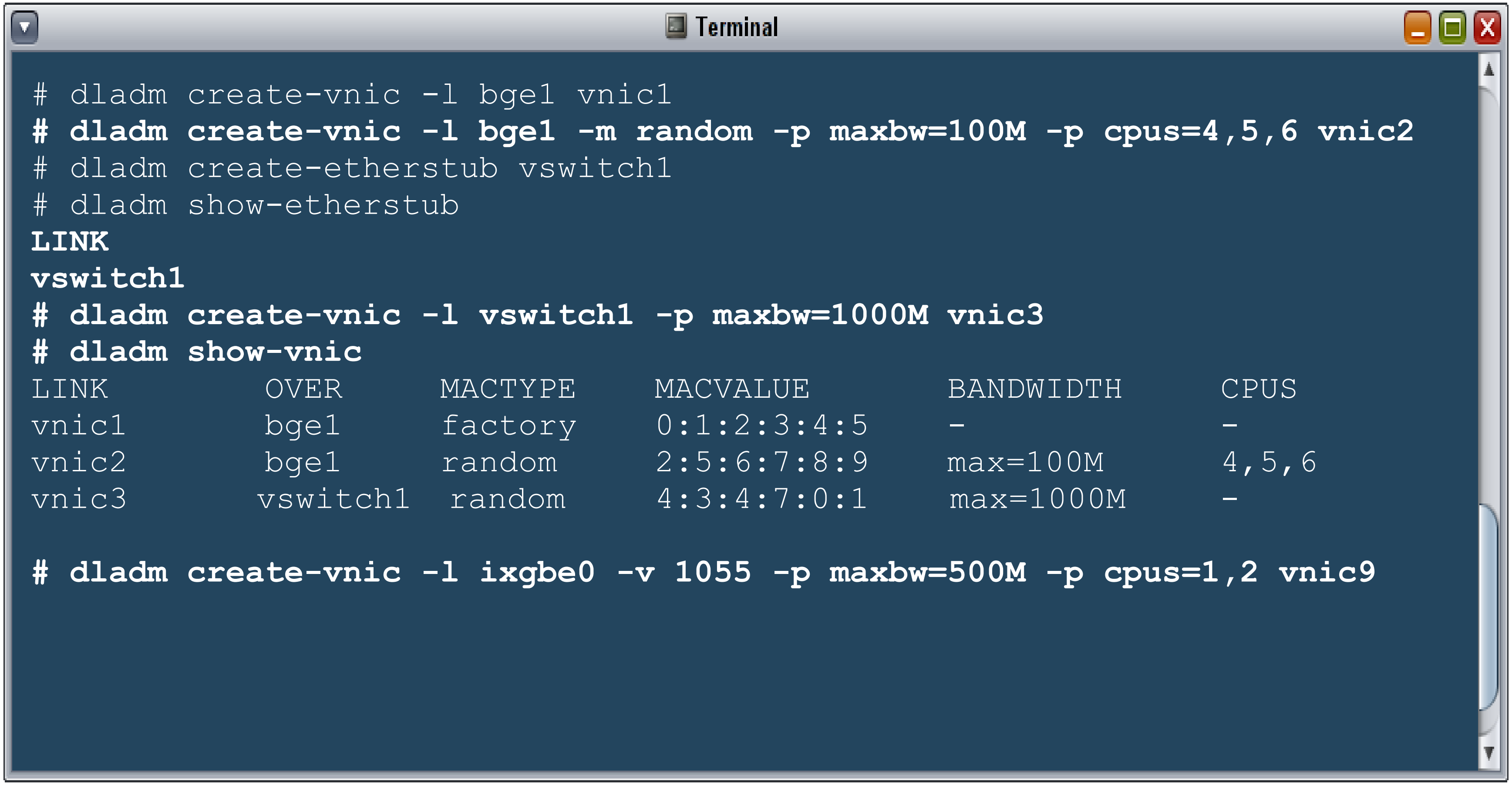
Crossbow Virtual Switching

- A virtual switch is created implicitly each time >2 VNICs are created on a data link
- The MAC layer provides packet switching semantics equivalent to an ethernet switch
  - Data path between VNICs created on top of the same data link
  - Connectivity between VNICs and physical network
  - Per VLAN broadcast domain, isolation between VLANs
- VNICs can be created on etherstub to create virtual switches independent from hardware

Crossbow Virtual Switching Example



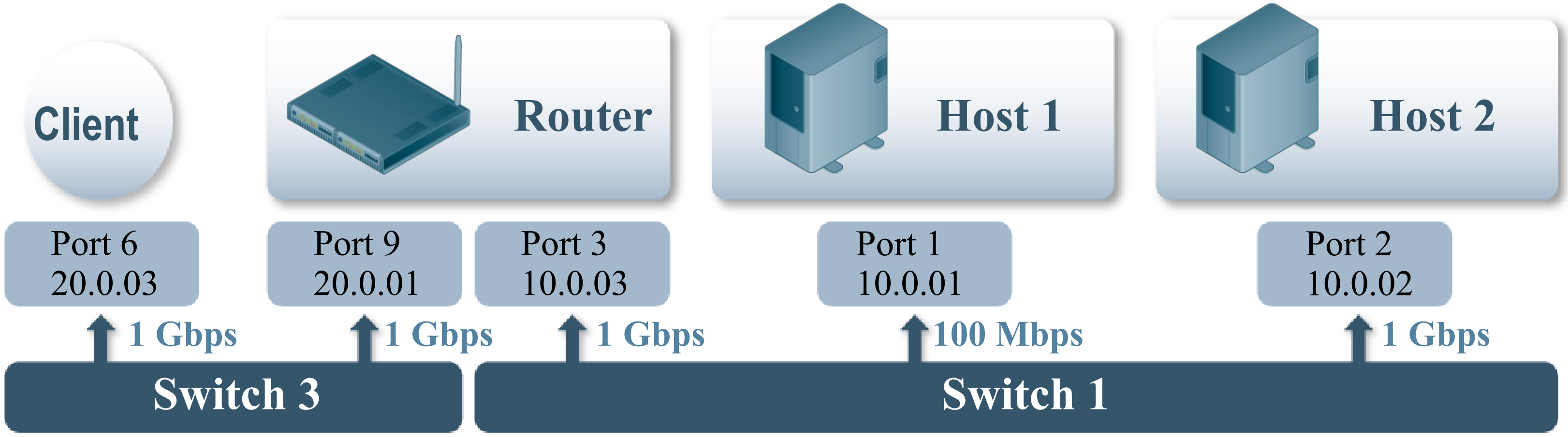
Virtual NIC & Virtual Switch Usage



```
# dladm create-vnic -l bge1 vnic1
# dladm create-vnic -l bge1 -m random -p maxbw=100M -p cpus=4,5,6 vnic2
# dladm create-etherstub vswitch1
# dladm show-etherstub
LINK
vswitch1
# dladm create-vnic -l vswitch1 -p maxbw=1000M vnic3
# dladm show-vnic
LINK         OVER      MACTYPE    MACVALUE       BANDWIDTH     CPUS
vnic1        bge1      factory    0:1:2:3:4:5    -             -
vnic2        bge1      random     2:5:6:7:8:9    max=100M      4,5,6
vnic3       vswitch1    random    4:3:4:7:0:1    max=1000M     -

# dladm create-vnic -l ixgbe0 -v 1055 -p maxbw=500M -p cpus=1,2 vnic9
```

Physical Wire w/Physical Machines



Client

Router

Host 1

Host 2

Port 6
20.0.03

Port 9
20.0.01

Port 3
10.0.03

Port 1
10.0.01

Port 2
10.0.02

1 Gbps

1 Gbps

1 Gbps

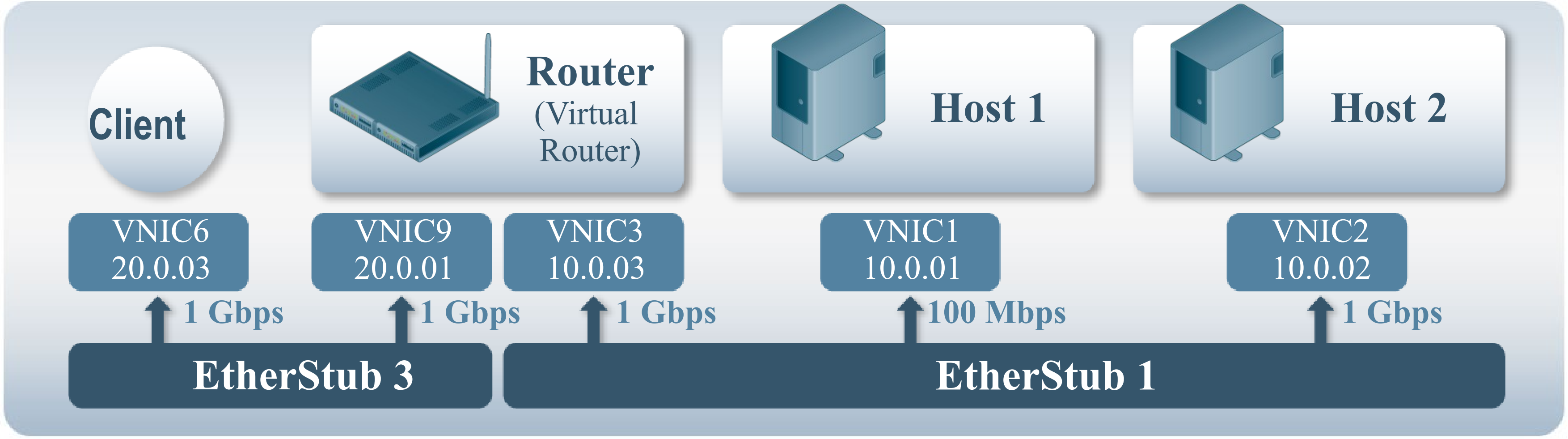
100 Mbps

1 Gbps

Switch 3

Switch 1

Virtual Wire w/Virtual Network Machines



Client

Router
(Virtual
Router)

Host 1

Host 2

VNIC6
20.0.03

VNIC9
20.0.01

VNIC3
10.0.03

VNIC1
10.0.01

VNIC2
10.0.02

1 Gbps

1 Gbps

1 Gbps

100 Mbps

1 Gbps

EtherStub 3

EtherStub 1

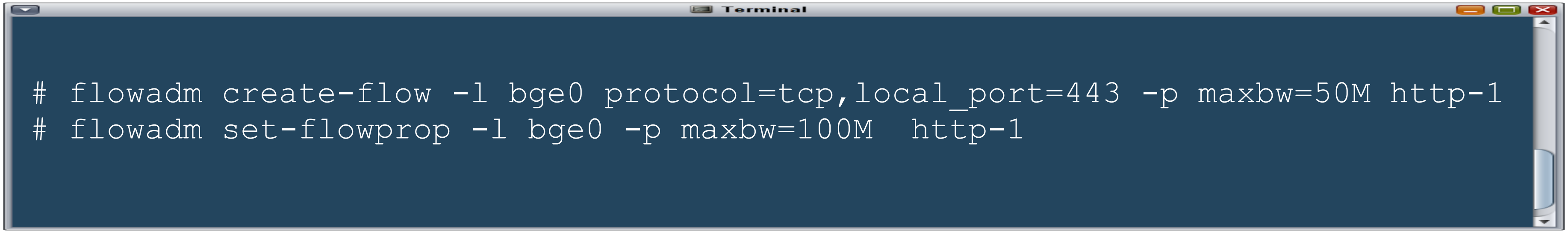
Virtual Network Machines

- A Virtual Network Machine (VNM) is a Zone or Virtual Machine associated with a set of network functions (routing, firewall, load balancing, etc)
- A VNM has dedicated VNIC(s) with configured link speed, CPUs
- Multiple VNMs can run on a single host, connected through virtual private networks (etherstubs) or to the physical network
- Use for simulation, consolidation, testing, etc

Crossbow Flows

- Crossbow **flows** based on the following attributes
  - > Services (protocol + remote/local ports)
  - > Transport (TCP, UDP, SCTP, iSCSI, etc)
  - > IP addresses and IP subnets
  - > DSCP labels

- The following properties can be set on each flow
  - > Bandwidth limits
  - > Priorities
  - > CPUs



```
# flowadm create-flow -l bge0 protocol=tcp,local_port=443 -p maxbw=50M http-1
# flowadm set-flowprop -l bge0 -p maxbw=100M  http-1
```

Join Us...

- Beer @ Crossbow and Solaris Networking BoF
  - > Tonight 10:30-11:30pm (Dover A&B)
  - > Presentation by Ben Rockwood (Joyent)
  - > vWire demo and deep-dive discussions
- OpenSolaris project and community
  - > http://www.opensolaris.org/os/project/crossbow
  - > crossbow-discuss@opensolaris.org
  - > networking-discuss@opensolaris.org

Crossbow Virtual Wire: Network In a Box

Nicolas Droux
nicolas.droux@sun.com
Solaris Kernel Networking