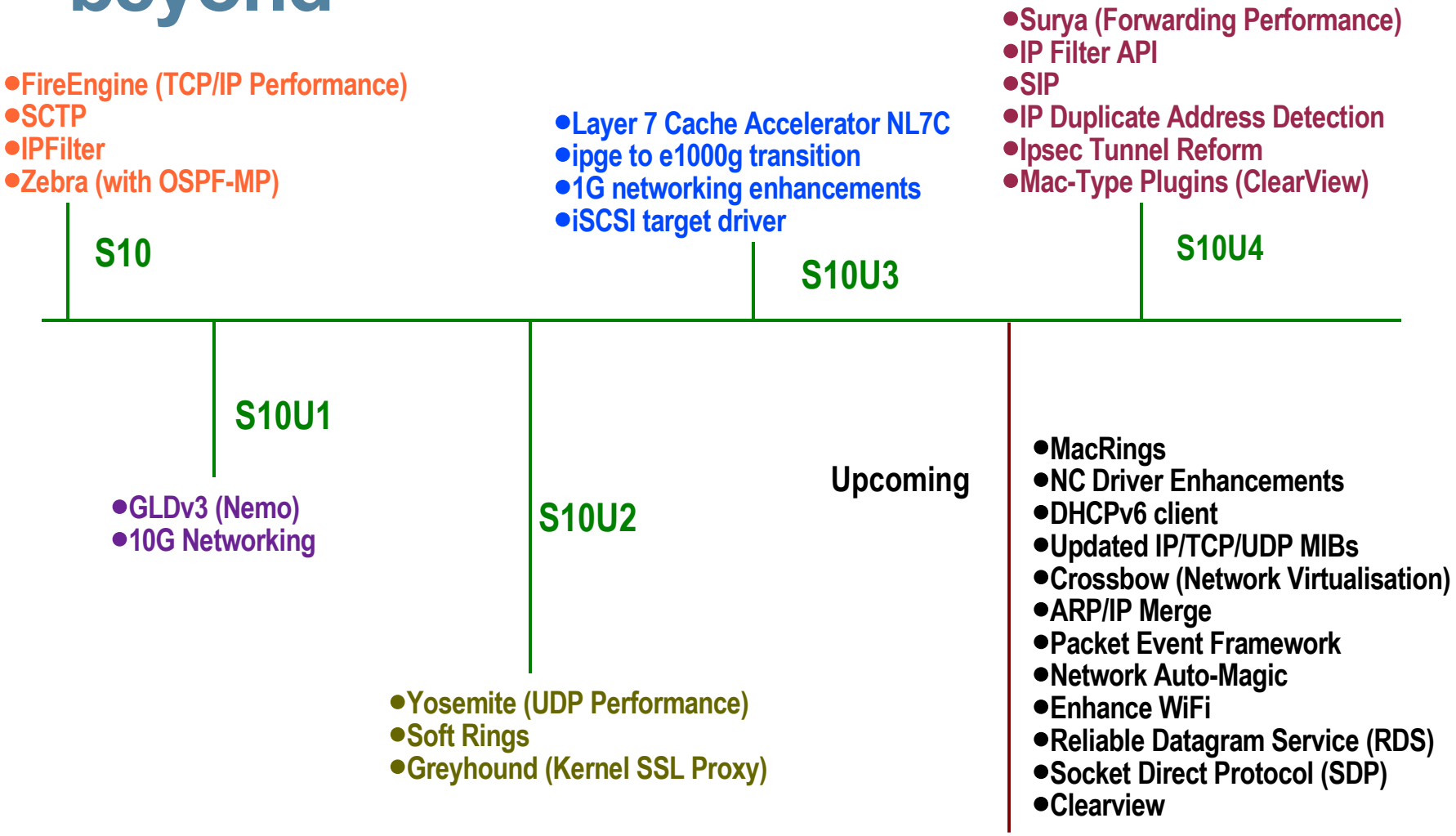
SOLARIS KERNEL NETWORKING

Darren Reed
Staff Engineer
Solaris Core Networking Technologies
Sun Microsystems, Inc.

Solaris Networking: Solaris 10 and beyond



Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - SIP
  - IPFilter
  - Yosemite
  - Surya
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances
  - IP Observability

FireEngine - Introduction

- The key workloads customers care about are SPEC99 (web), SPEC99-SSL (secure web), SPECjAppServer (j2ee), and their own workloads
- Solaris 10 has a cutting edge networking architecture (FireEngine) that makes it competitive on low end and linearly scale on mid/high end
- Goal is to “Move more data – faster, cheaper and more securely!”

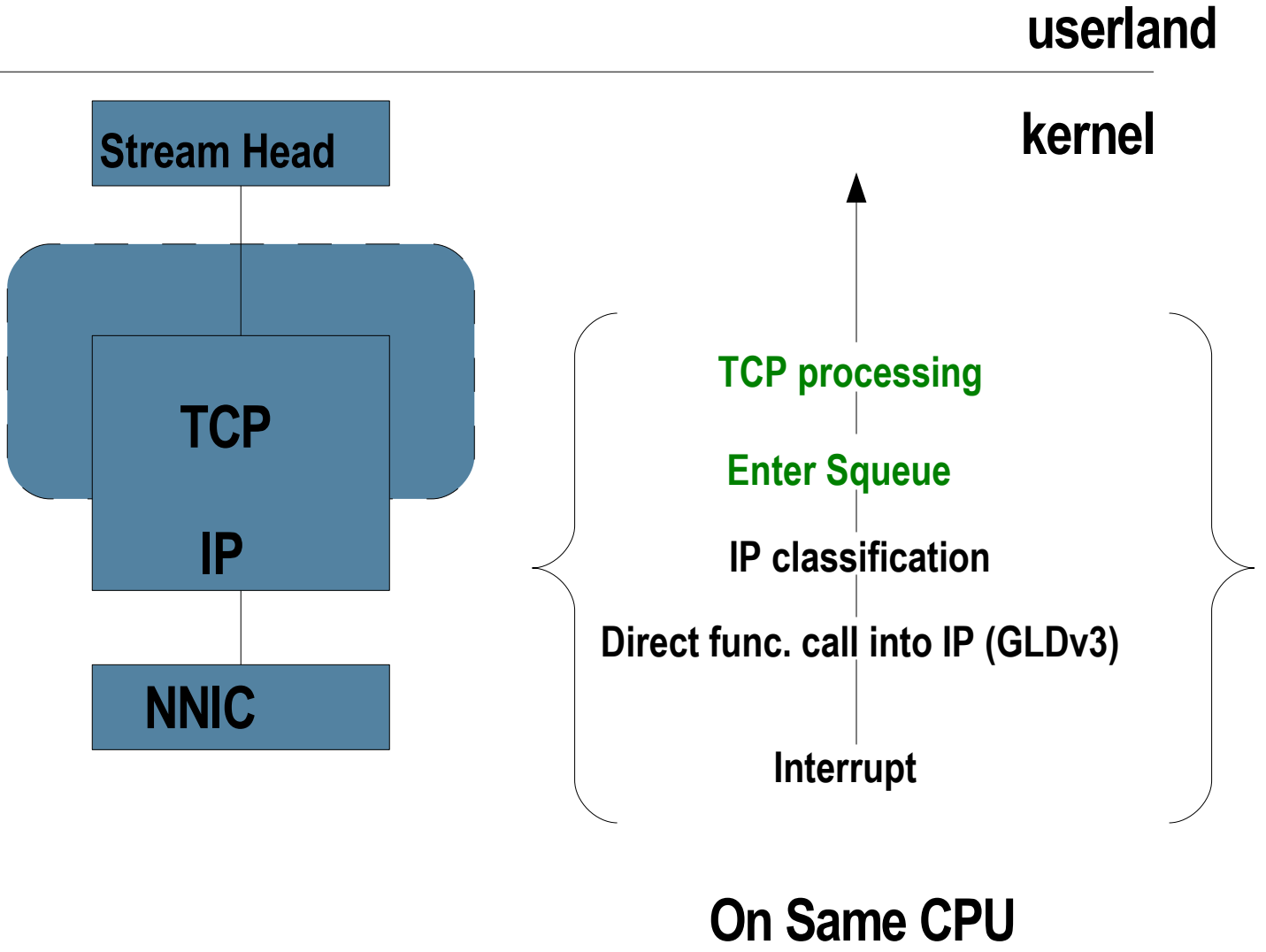
FireEngine – TCP/IP performance

- Prior to Solaris10, networking stack was STREAMS based:
  - context switching
  - poor cache locality
- Performance improvements:
  - Web based workloads: 30 – 45% gain
  - Bulk data transfer: 20 – 30% gain
  - TTCP, netperf: 35% gain

FireEngine – Goals

- Networking Performance
  - Improve per-packet latencies
  - Achieve linear scalability
  - Cut per-packet processing cost
- Out of the box performance
- Feature growth management
- Stack sustainability and observability

FireEngine (TCP/IP performance)



FireEngine – Design Principles

- Reduce the per packet latencies by cutting the number of queues a packet goes through
- Improve data locality by cutting the interaction between CPUs and improving data locality
- Reduce per packet processing cost by improving interaction between various layers (i.e. Socket, TCP, IP, data link, etc)

FireEngine – Vertical Perimeter (squeues)

- Squeue is a per CPU common serialisation queue (FIFO) for all inbound/outbound packets
- Squeue provides the mutual exclusion to all TCP connnections without locks (lockless design) by allowing only one thread to process it at any given time
- Packet once picked up for processing is taken all the way to socket (on inbound) or NIC (on outbound) giving it the property of Vertical perimeter

FireEngine – IP Classifier

- Use a connection classifier early in IP for incoming packet
- The connection structure ('connp') contains all the necessary information:
  - The CPU/squeue the packet needs to be processed on
  - The string of functions necessary to process the packet (event lists)

FireEngine – Squeue + Classifier

- Create a per CPU squeue index on CPUid
- Bind a connection to a particular squeue so packets for that connection are always processed on the same squeue
- Bind each inbound connection to the squeue attached to the interrupted CPU for incoming connection to maintain data locality and vertical separation
- Use the classifier to direct packets to the CPU they need to be processed on

FireEngine – TCP/IP merge

- Use function calls between TCP and IP to reduce per packet processing cost
- Separate and optimize hot paths
- Merge TCP/IP in one STREAM module (fully MT)
- The STREAM entry points are manipulated based on whether someone opens /dev/tcp or /dev/ip
- TCP/IP modules behaves the same – if someone opens /dev/ip they see IP behaviour and /dev/tcp for TCP behaviour

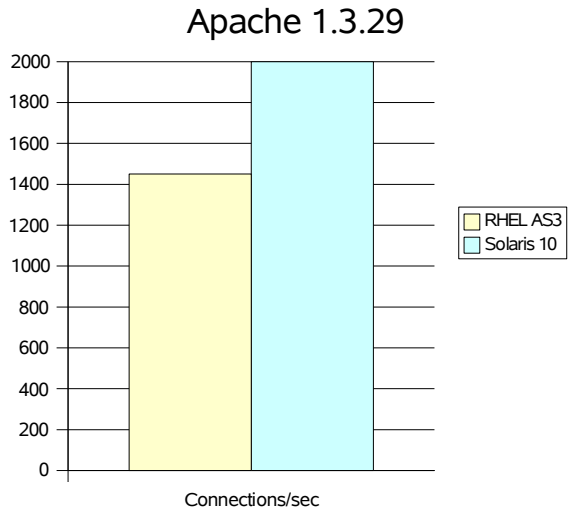
FireEngine – Current status

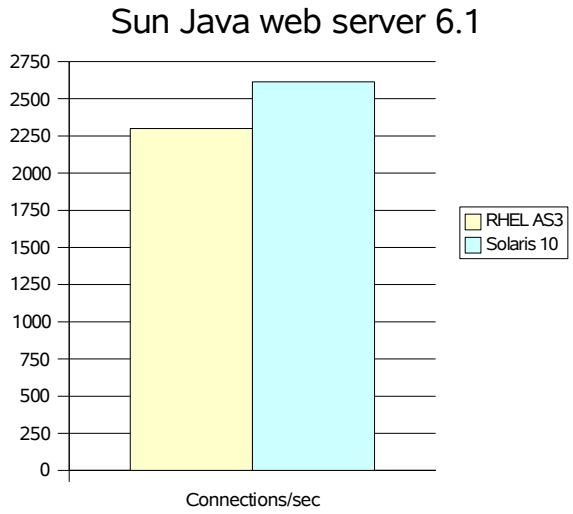
- FireEngine Phase 1 is now available in Solaris 10
- Web-like workload performance gains
  - 45% SPARC
  - 43% x86
- Other gains:
  - 10% SSL
  - 20-40% throughput (ttcp)
- On a v20z, Solaris is ~10% faster than Linux using Apache/Sun One Web for web based workload

FireEngine – Current status (cont.)

- Solaris 10 – Webbench:
  - Static – Outperforms Windows 2003 by 26%
  - Dynamic – Outperforms Windows 2003 by 29%, RHEL-AS3 by 3%
  - E-Commerce – Outperforms Windows 2003 by 18% and RHEL-AS3 by 14%
- Solaris 10 – NICs:
  - Saturates a 1Gb link using 1x2.2GHz Opteron with only 8% CPU
  - Drives a 10Gb link at full PCI-X speed (7.3 Gbps) with 2x2.2GHz Opteron at under 50% CPU

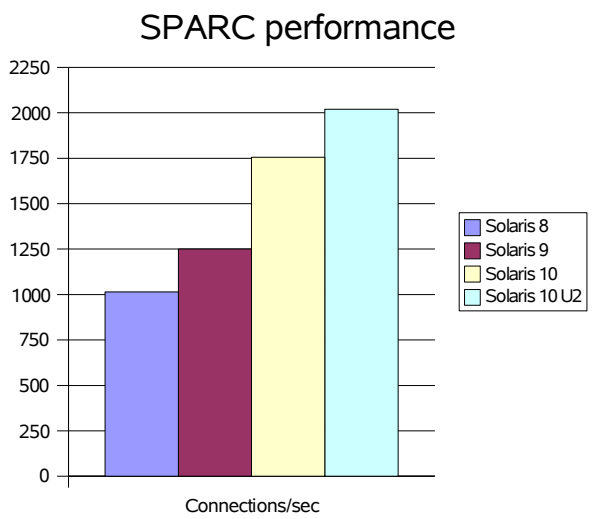
Solaris web performance vs RedHat





- Configuration
  - X86: v20z (2x2.2GHz, 6Gb RAM, 2x1Gb NICs)
- Connections/sec are number of connections that can be handled at a certain bit rate (similar to SPECweb99 conn/sec)

Solaris web performance



- Configuration
  - X86: v20z (2x2.2GHz, 6Gb RAM, 2x1Gb NICs, Zeus 4.1r4)
  - SPARC: Sunblade 2500 (2x1.2GHz USIII+, 8Gb RAM, 2x1Gb NIC, Zeus 4.1r4)
- Connections/sec are number of connections that can be handled at a certain bit rate (similar to SPECweb99 conn/sec)
- S10U2 numbers are based on project Nemo

FireEngine – Architectural Advantages

- IP Classifier
  - Allows FireEngine to bind connections to CPU, giving much better data locality
  - Coupled with vertical perimeter gives linear scalability
- Vertical Perimeters
  - One common outbound/inbound queue
  - Allows a packet to traverse through the stack without getting queued in multiple places
  - Offers very tightly bound latencies even under heavy load

FireEngine – Architectural Advantages (cont.)

- Reduced per packet processing cost
  - Lockless design cuts mutex contention
  - Per CPU data structures & MIBs reduce cache misses and cross calls
  - Improved interaction between layers (socket, TCP, IP, NIC, etc) allowing optimized fast paths
- Designed for future technologies
  - Dynamically switch between polling and interrupts based on CPU backlog, etc.
  - Offload technologies: TOE, SSL
  - RDMA

FireEngine – Further reading

- http://www.sun.com/2004-1012/feature
- http://www.sun.com/software/solaris/networking.jsp
- http://www.sun.com/software/solaris/performance.jsp
- http://www.sun.com/bigadmin/xperts/sessions/11_fireengine/

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - SIP
  - IPFilter
  - Yosemite
  - Surya
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

Stream Control Transmission Protocol (SCTP)

- Transport Protocol supports functions critical for telephony signalling
  - Multi-streaming
  - Fault tolerance at the transport layer
- Applications can access SCTP using SOCK_STREAM (one-to-one) or SOCK_SEQPACKET (one-to-many)
- RFC's implemented: 2960, 3309, 3758, 3873, 4460
  - #include <netinet/sctp.h>
  - sctp(3LIB) -lsctp

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - SIP
  - IPFilter
  - Yosemite
  - Surya
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

Session Initiation Protocol (SIP)

- Signalling protocol for multimedia applications
- C API provided
- SIP stack written in C as a userland library
- Consists of {Header, Transaction, Dialog} Mangement and Message Formating Layers
- Goal is to allow user to build SIP applications
- Solaris ships SIP proxy server “SER” from iptel
  - Open source implementation, shipped in /usr/sfw
  - Can be used as a registrar/proxy/redirect server.

SIP – Standards based implementation

- RFC based implementation
  - 3261 – SIP specification
  - 3262 – Reliability of Provisional Responses in SIP
  - 3265 – SIP Event Notification
  - 3323 – Privacy Mechanism for SIP
  - 3325 – Private Extensions for Asserted Identity in Trusted Networks
- SIP C library for developers
  - -lsip
  - #include <sip.h>

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - SIP
  - IPFilter
  - Yosemite
  - Surya
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

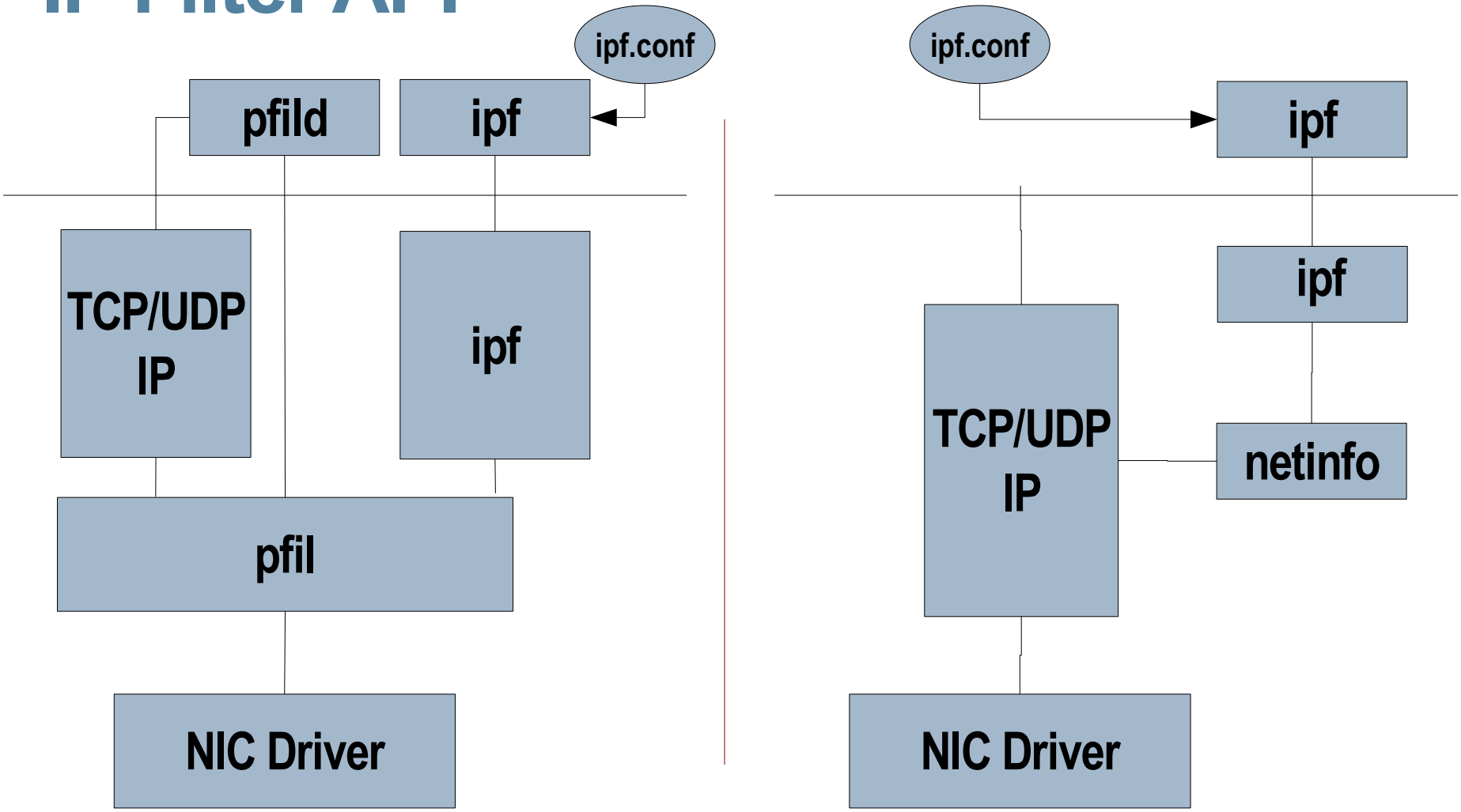
IPFilter – Introduction

- Firewall/NAT capabilities for Solaris
  - Open source development, Sun support
  - IPv4 firewall/NAT
  - IPv6 firewall
- STREAMS module approach
- Host or Gateway filtering
- API to support transparent proxies

IPFilter – Current Work

- APIs to remove reliance on STREAMS
  - Performance benefit 20-30%
  - Access to machine-local traffic
    - Inter-zone, Intra-zone, loopback
- Updating IPFilter version in Solaris
  - Last bump from 4.0.3 to 4.1.9
  - Ongoing...

IP Filter API



Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

Yosemite (UDP Performance)

- Merge UDP/IP
- Checksum offload
- Udpsockfs- direct path from sockfs to udp
- MDT for fragmented UDP
  - reduced IOMMU cost
- Leverage Nemo/FireEngine
  - squeues, better flow control
- TCP loopback fusion
  - short curcuit end points

Yosemite – The Results

- 20% Improved Latency
- 12-170% libMicro Improvements
- 20-90% TIBCO Rendezvous
- 15-20% VolanoMark
- 35-65% (frag.) TX Throughput
- 25-110% (frag.) RX Throughput
- 25% NFS3/UDP Read Throughput

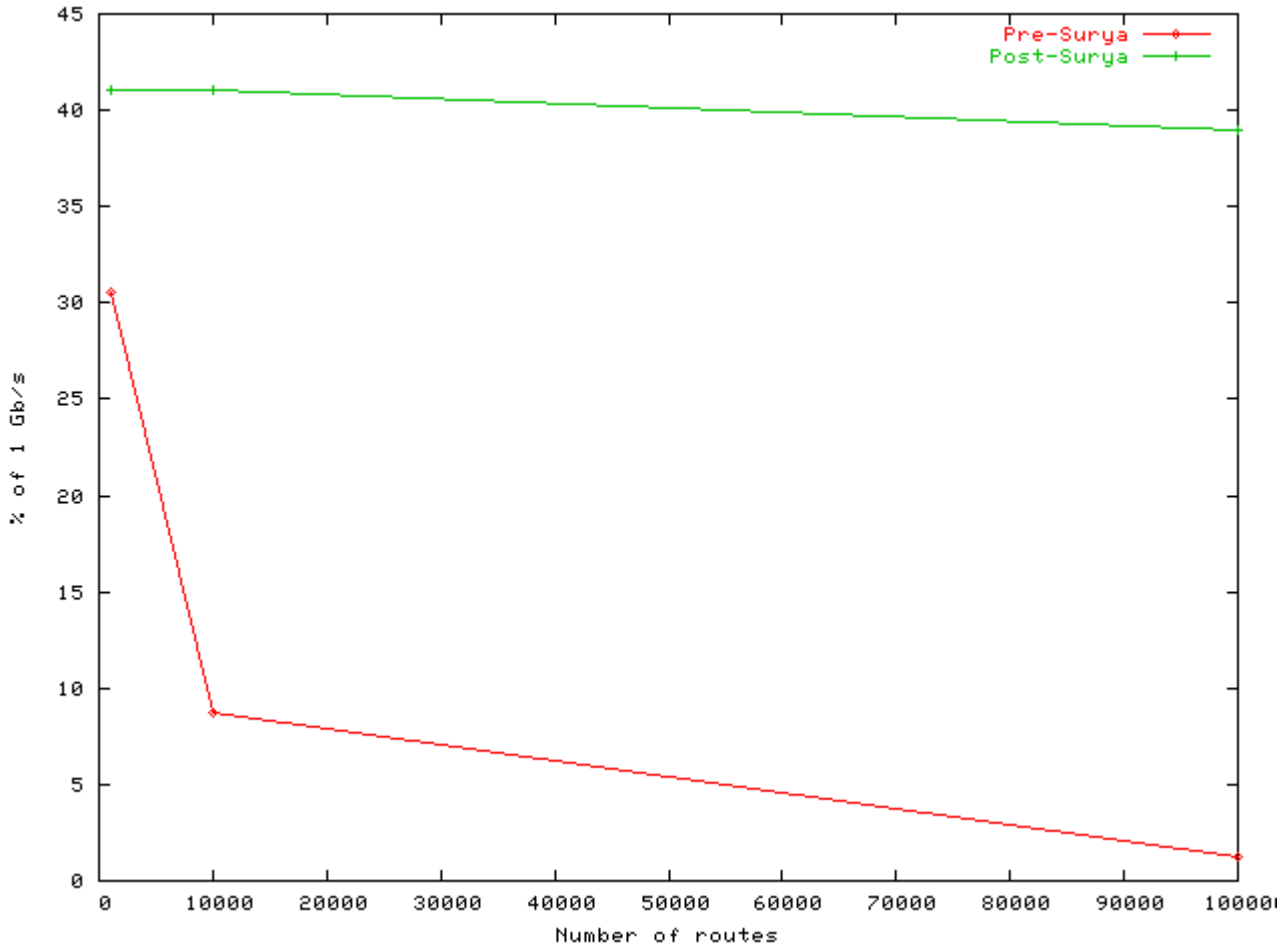
Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

Surya – IPv4 Forwarding improvement

- Forwarding throughput and scalability improvement
- Optimized forwarding path logic
- New routing table lookup logic (PATRICIA)
  - Using the BSD implementation

Surya (IP Forwarding improvement)



Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

Updated MIB Support

- IP/TCP/UDP MIB support
  - Make Solaris compliant with latest MIB RFCs:
    - 4022 (TCP)
    - 4113 (UDP)
    - 4293 (IP)
- Remain backward compatible and provide all information to SNMP

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

NEMO

- GLDv3
- Interrupt Moderation
- Trunking
- Extensions

Nemo – GLDv3

- GLD – Generic LAN Driver
  - Evolving interface:
    - Version 1 in Solaris 7
    - Version 2 in Solaris 9
  - Current drivers: bge, nge, xge, rge, ixgb, e1000g
- High performance framework that IP can use to control all activities/resources without burdening the device driver writer.
- Work in progress to make the API robust and "*future proof*"

Nemo: Interrupt Moderation

- Networking interrupts are bad because writers gets pinned, context switches, etc.
- Bind a NIC to a Squeue and the let the Squeue own the NIC and have the ability to turn off interrupts
- If Squeue becomes backlogged, it turns the NIC interrupts off
- More CPU is available to process backlog
- The packets that were received by the NIC while interrupts were disabled are sent up as a chain as soon as interrupts resume.

Nemo: Interrupt Moderation (cont.)

- Expect another 20% improvement web workloads
- Sample mpstat output:

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
```

- Notice the decrease in interrupts, context switches, mutex contentions, etc. and increase in idle time

Nemo (GLDv3) – Performance improvements

- Significant network performance improvement with better interrupt management and streamlined code paths
- 25% improvement on x86 and 20% on SPARC platforms on web workloads
- Sample mpstat output:
- ***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
```

- Decrease in context switches, mutex contentions etc and increase in idle time.

Nemo – Trunking

- Create trunks (link aggregations) of 1Gb NICs or 10GB NICs
- IEEE 802.3ad compliant, including LACP (Link Aggregation Protocol)
- Each member or the trunk is owned by individual Squeues which control the rate of arrival of packets
- No specific support needed from device driver
- Close to linear scalability for a trunk of 4 1Gb NICs

Nemo – VLANs

- Industry standard VLAN support enables greater network configuration and management capabilities

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

IPv6 Certification

- Undertaken with the University of New Hampshire
  http://www.iol.unh.edu/services/testing/ipv6/
  - > Phase I Complete
  - > Phase II Complete
- SunOS 5.11 build 52

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

CrossBow – Network Virtualization

- Support for dumb NICs through 'soft rings' and NEMO unification
- Supports guest OS stacks via Xen (i.e complementary to Hard Virtualization)
- Management through dladm, netrcm, zonecfg (assign b/w, priority)
- Accounting (per service, protocol stats) and capacity planning now possible

Crossbow – Introduction

- Problem Statement
- Goals
- Architecture

Crossbow – Problem statement

- Financial Services
  - Trading house starts offering free financial information to attract customers
  - Brokerage customers start complaining that trading site slows down
  - The paying customers start deserting
- Large ISP
  - ISP wants to deploy virtual systems on the same physical machines
  - ISP sells each virtual system at different price levels to its customers
  - Any virtual instance can overwhelmed the shared networking resource
- What happened?
  - Critical services are overwhelmed by non-critical services, traffic types or virtual systems.
  - No usable mechanism available for fairness, priority and resource control for networking bandwidth

Crossbow – Future networking stack

- The network stack of tomorrow needs:
  - **Fairness**: The ability to allow various types of traffic to share the bandwidth and associated compute resources in a fair manner.
  - **Priority**: The ability to prioritise traffic or service types.
  - **Resource Control**: The ability to manage network bandwidth and allocate networking resources.
- Customers demand these features integrated as part of architecture and without performance penalty.
- Critical to achieving true utility computing, network virtualisation, etc.

SoftRings (Crossbow Phase 0)

- Motivation: CPU speeds not keeping with networking bandwidth
- Soft Rings are software queues (available in Nemo NICs) that allow the inbound load to spread across other CPUs
- Interrupt thread queues packet onto the soft ring
- Worker thread woken up and processes packet
- On niagara sytems, on by default (20-40% perf improvement- for TCP workloads)
- Turned on by setting ip_squeue_fanout = 1

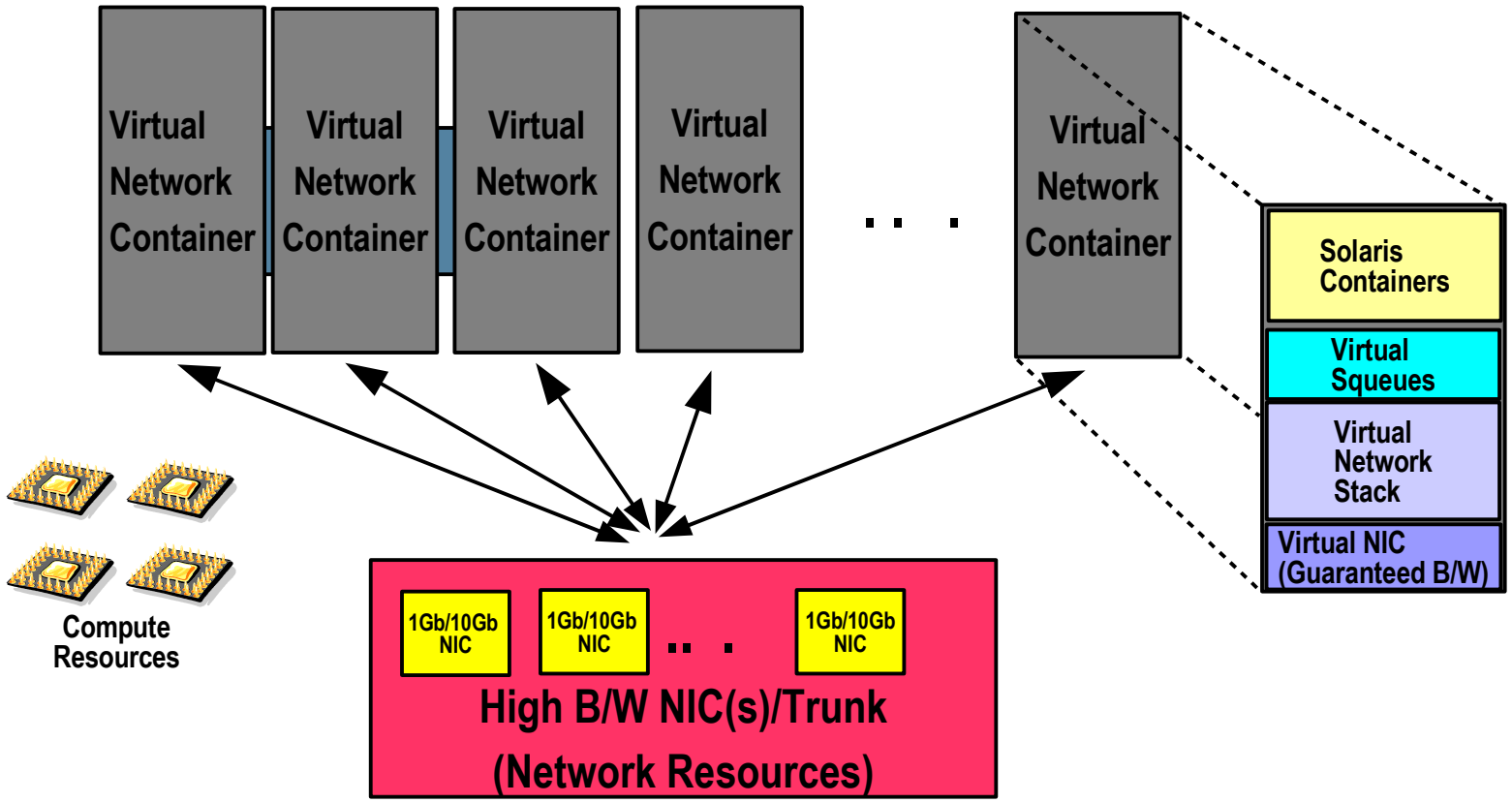
Network Virtualization

- Virtualize the 1Gb and 10Gb NICs based on traffic type (protocol, service, container)
- Control the bandwidth utilization and CPU utilization due to networking for each virtual NIC
- Each Virtual stack instance should be able to treat the virtual NIC assigned to it as real and apply stack specific resource control

Technical Obstacles

- Obstacles to achieving network virtualization:
  - Network processing in interrupt context
  - Anonymous packet processing in kernel
  - Common queues
- Performance can be degraded by the extra processing to enforce fairness, resource control or network virtualization

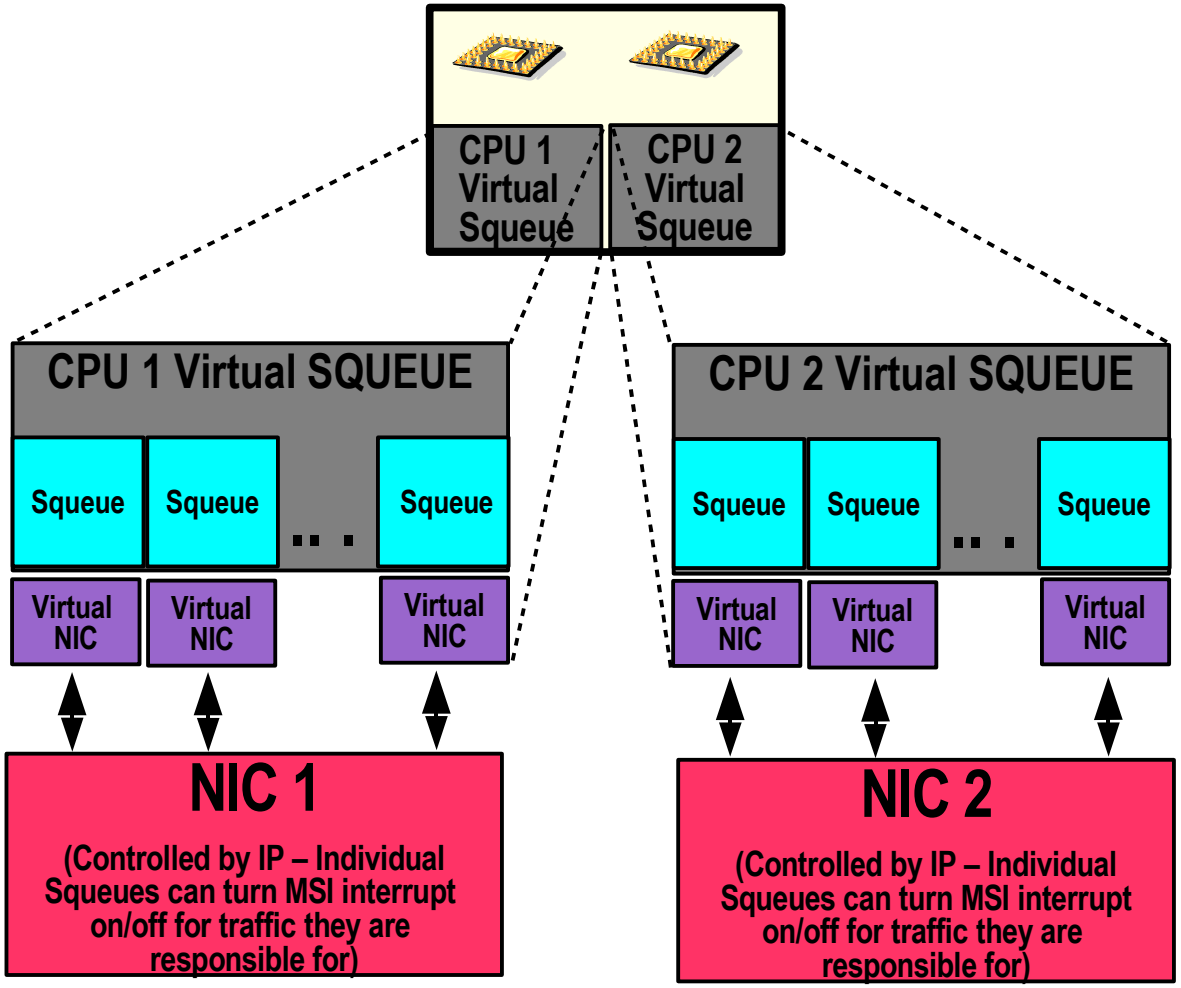
Virtual Network Stack



The Crossbow Architecture

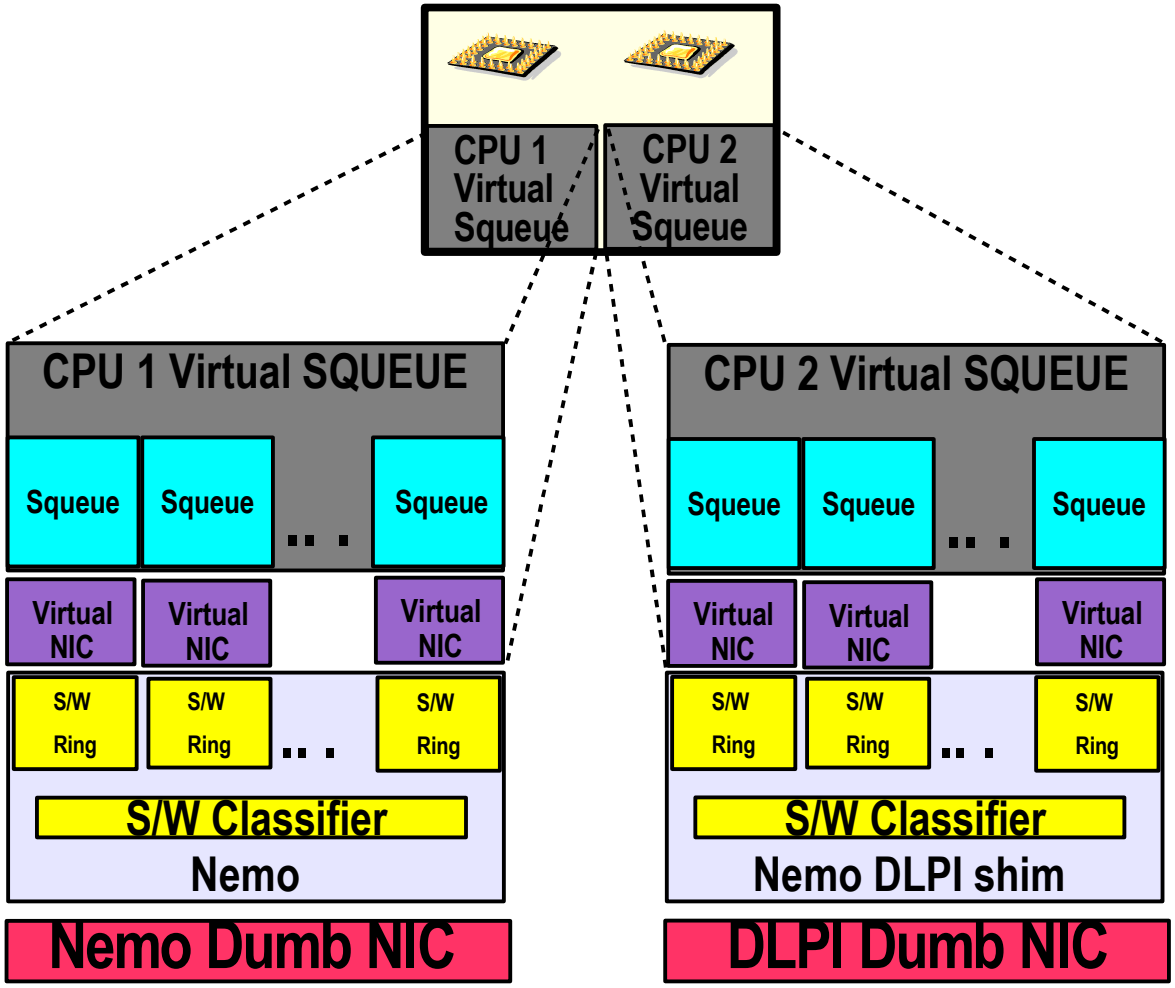
- Use the NIC to separate out the incoming traffic per virtual stack and use a per stack MSI interrupt
- Traffic for each virtual stack is stored on the NIC buffers itself till retrieve by the squeue
- The FireEngine Squeue controls the rate of packet arrival into the virtual stack by dynamically switching between interrupt & polling
- Incoming B/W is controlled by pulling only the allowed number of packets per second
- Virtual stack priority is controlled by the squeue thread which does the Rx/Tx processing

Virtual Stacks



The Squeue controls the rate of packet arrival into the system for their virtual stack

Virtual Stacks with Dumb NICs

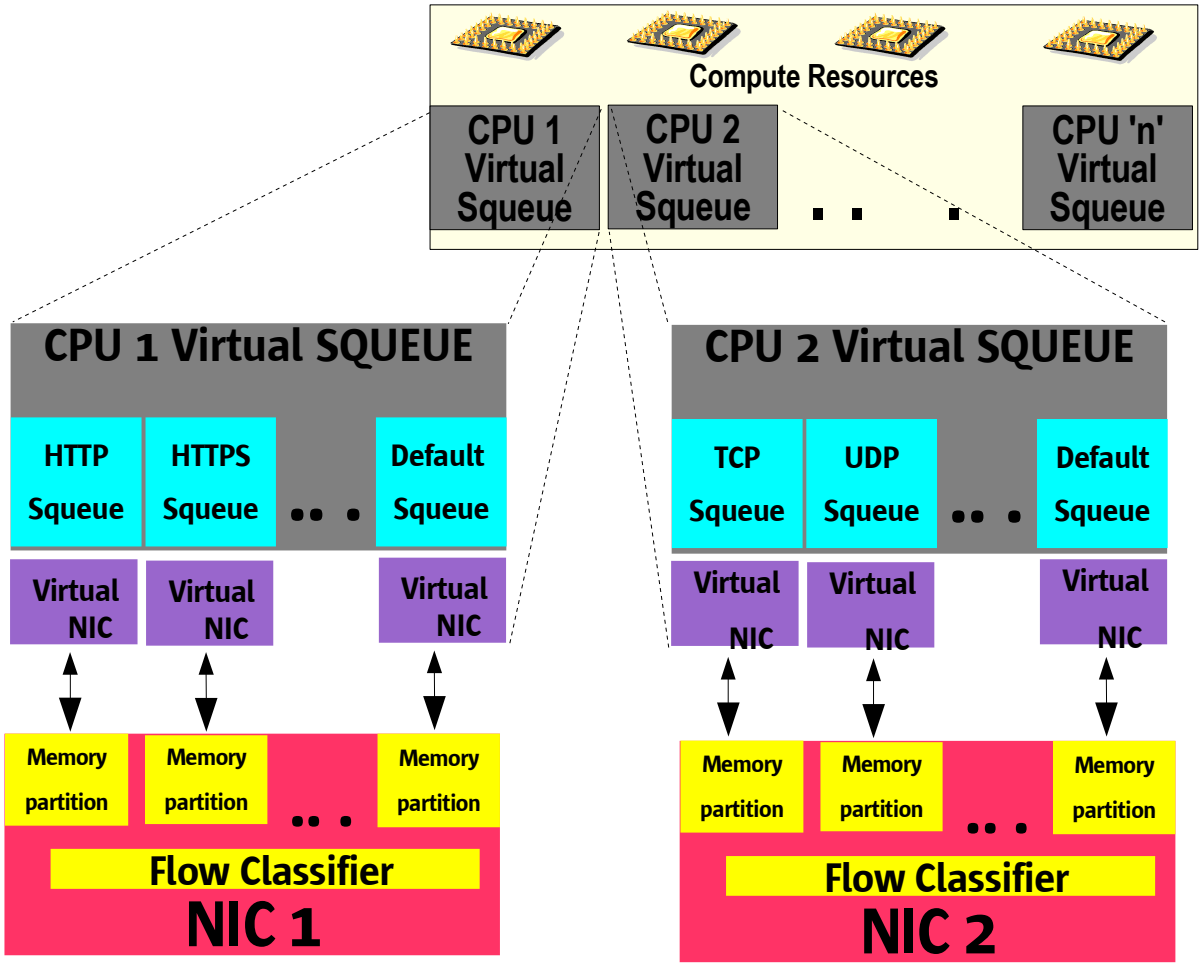


The Squeue switches the Rx Ring between interrupt and polling mode and controls the rate of packet arrival

Virtual Stack per Service / Protocol

- Virtual stacks can be created around the few important services (like HTTP/HTTPS) with individual resource control (priority, B/W usage, components in the stack like firewall, IPsec etc)
- Virtual stacks can also be created around protocols with their individual resource control
- A default virtual stack is automatically created to deal with unspecified traffic

Crossbow (Virtual Stacks)

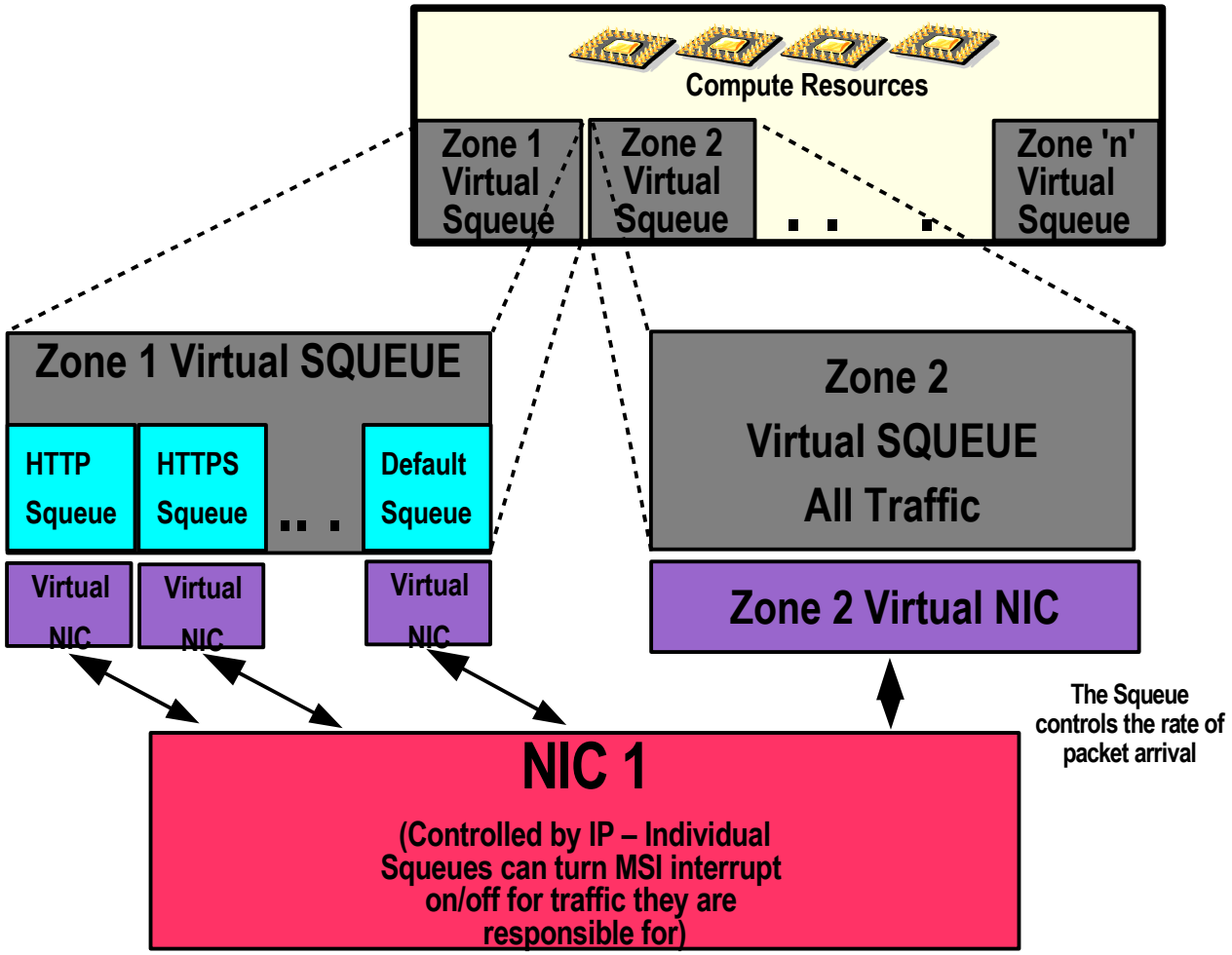


The Squeue switches the MSI interrupt per stack between interrupt and polling mode and controls the rate of packet arrival

Virtual Stack per container

- Each Solaris container can have its own virtual stack with private routing table etc.
- When container is created, the total resource and number of possible virtual stack with the container is specified
- The Container administrator can configure the allocated virtual stacks to its own taste
- Each Container can have its own routing table, firewall, etc and tune it according to its requirement

Virtual Stacks - Containers



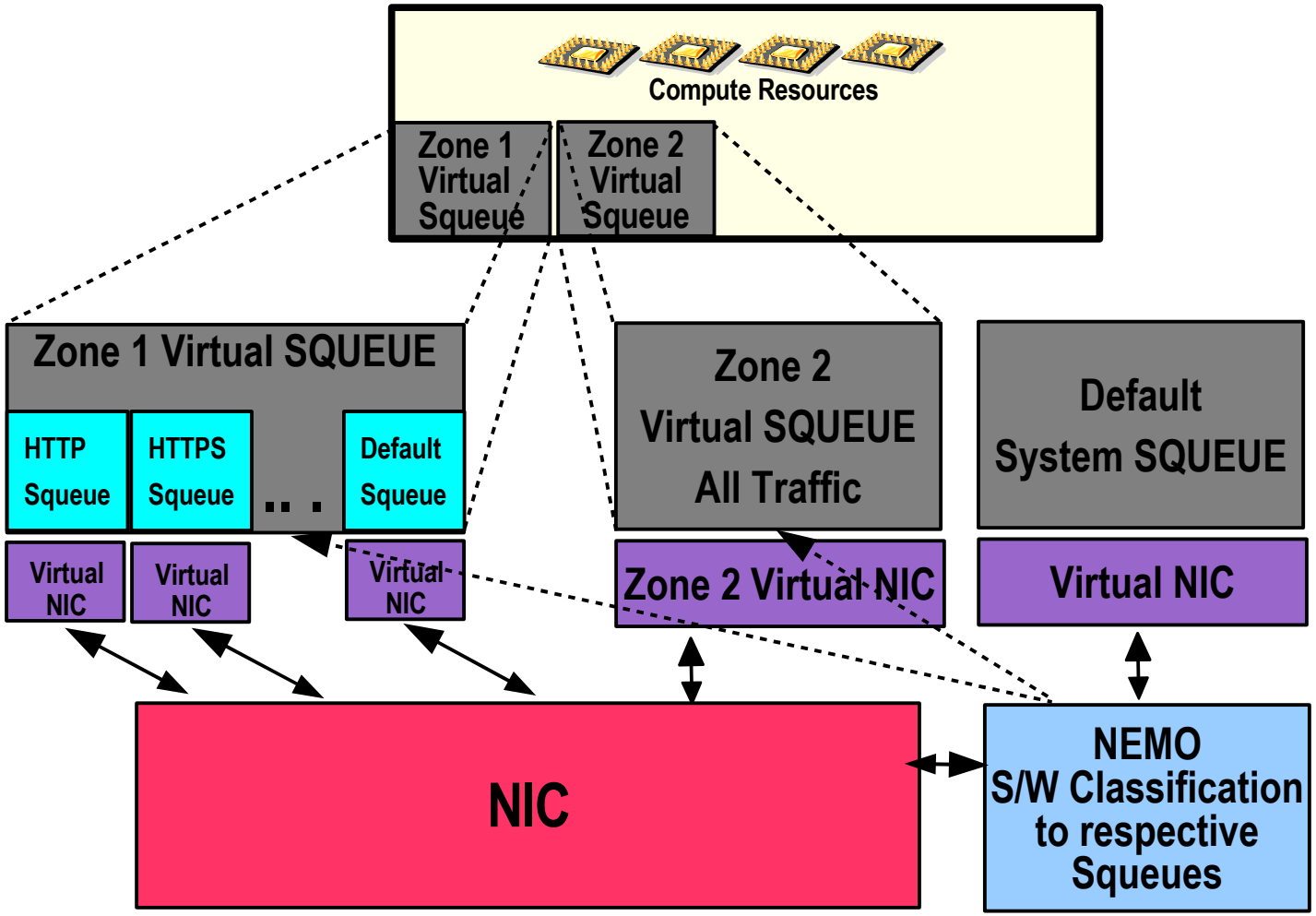
NICs

- Crossbow will support existing NIC today by separating out the flows in Nemo layer itself

Defense against DOS

- Denial of Service attack (DOS) are a threat today & have the ability to cripple the entire grids
- Dynamically lower the priority (and B/W) of the impacted service or container to minimize impact to other
- Under attack, impacted services start all new connections under lower priority (limited resource) stack
- Connections transitions to correct stack after authentication (or by means of other heuristics)

Defense against DOS attacks



Fair Accounting System

- Finer grain accounting comes for free
- We can now do per squeue or receive ring accounting to track usage by a container, service or protocol
- A user space daemon can pull the statistics out at fixed interval and do accounting etc.

Enterprise Network Appliances

- Pushing general purpose OS as routing platform into enterprise is difficult today due to performance and fairness issues
- Crossbow enabled Solaris makes the cut by creating flow based on traffic types which are dealt by their own virtual stack so one type can't impact other
- Coupled with offload capabilities, Solaris on a 2 CPU v20z can easily handle 6-7 Gbps of traffic

Value Proposition

- True utility computing
- Defense against DOS attacks
- Fairness, priority and resource control as part of the architecture
- Increased performance (out of box)
- True virtualization (per service, protocol, or container stack)
- Increased system utilization

CrossBow (Network Virtualization)

- Motivation: No usable model for fairness, priority and resource control for networking bandwidth
  - Crossbow Architecture:
    - Divide NIC resources (memory, DMA channels etc) to give Virtual NICs
    - Hardware classifier classifies flow based on specified criteria
    - FireEngine squeue controls packet arrival on each VNIC (by dynamically switching between interrupt/polling)
    - Virtual stacks (IP instances) with 'bundle' of squeues/VNICs; per zones container possible

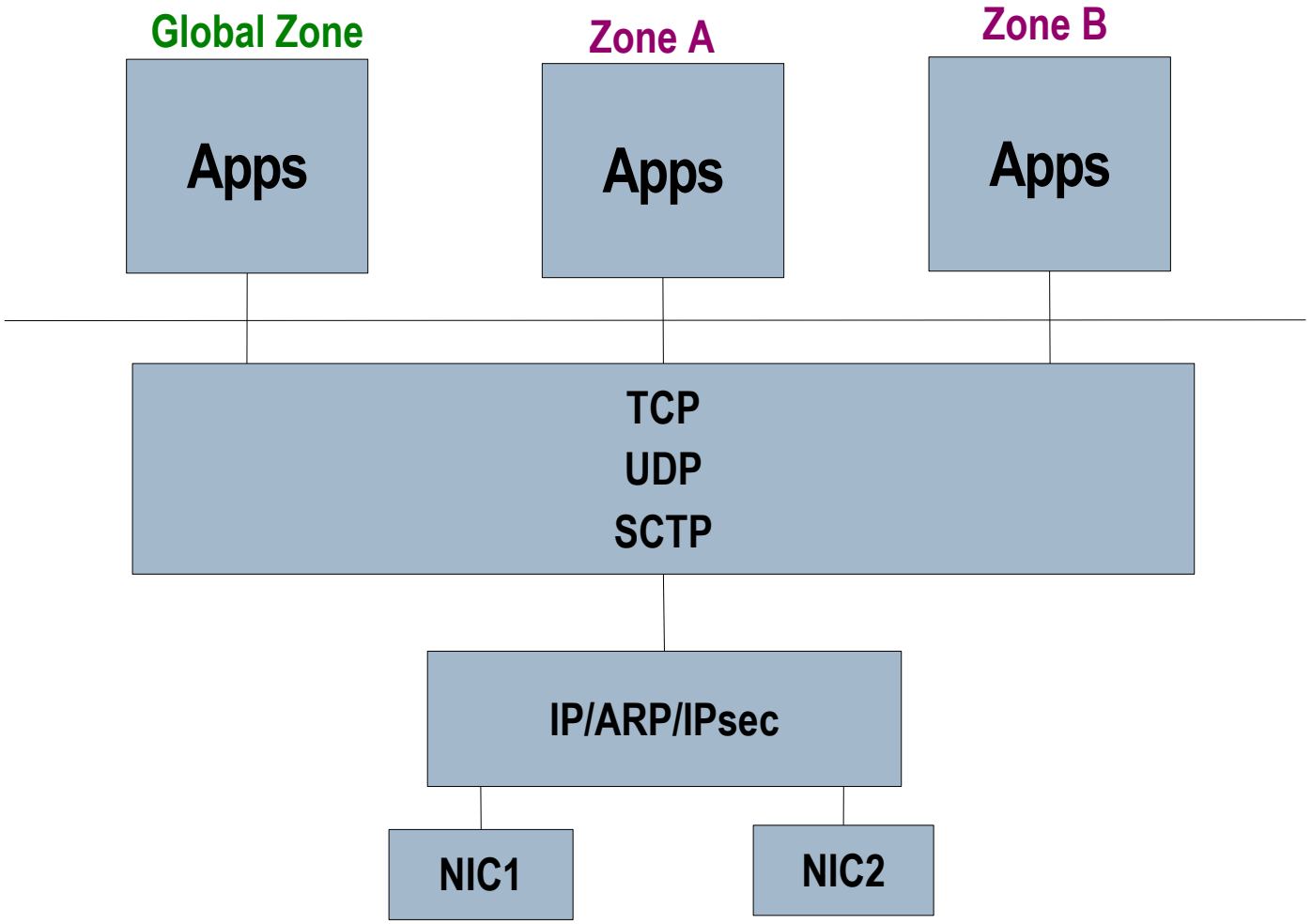
Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

IP Instances

- Zones Networking Today
- Aside – Zones Technology Brief
- Project Goals
- Shared vs Exclusive

Zones Networking – Today



Aside – Zones/containers introduction

- Virtualisation Technology
- Provides
  - Seperate process space
    - Inter-zone communication must be via TCP/IP
  - Seperate user space
  - Confined (i.e. chroot'd) disk space
- Global Zone and Local zones
  - Visibility of the system
- Zone administration
  - Subset of tasks required for Global Zone

Aside – Zones Networking

- Restricted visibility
  - Routing table = associated with network interface(s)
- Administered from the global zone
  - Cannot use DHCP
- Has its own socket port number space
  - Can listen on “0.0.0.0 port 80” in every zone
- Limited firewall capabilities

IP Instances – project goals

- Provide a more robust architecture for zones networking
  - Provides per-zone routing tables
  - Provides per-zone ARP table
- Allow a zone to have its own instance of IP

IP Instances – Shared vs Exclusive

- Shared stack
  - Uses global routing table
  - Global interface management, performance tuning, etc
  - Short-cut routing
  - Packet filtering applied to the entire machine
- Exclusive Instances
  - Routing table per zone
  - Each zone can tweak TCP, etc, settings in its own way
  - Each zone decides what filtering it wants

IP Instances – Delegation of control

- An exclusive instance isn't managable from the global zone
- Local zone root has full control over IP
  - Use of ndd to tune IP is allowed and is private
- Network interfaces delegated for exclusive use are not visible in the global zone

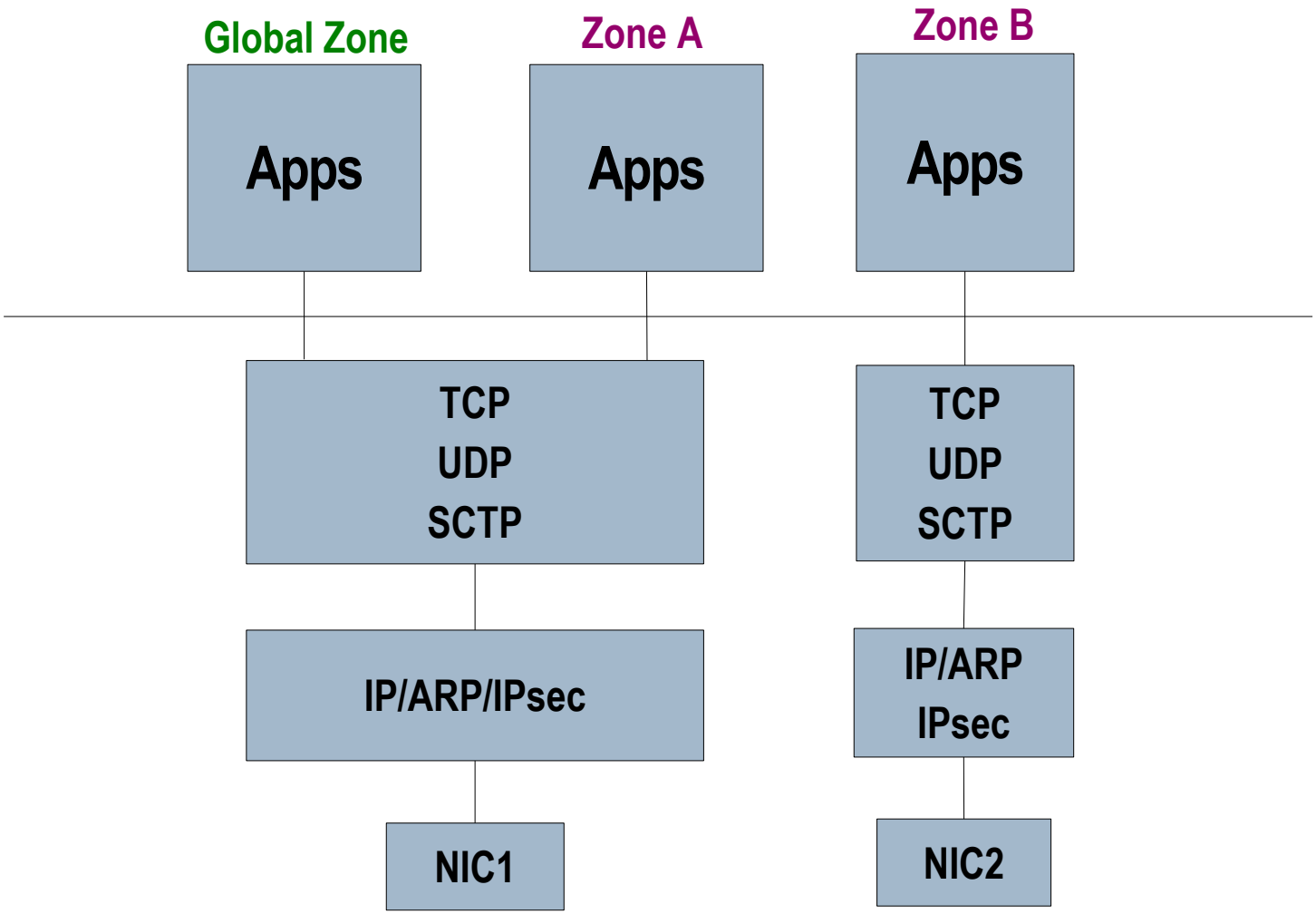
IP Instances – Changes to security

- Privilege split of SYS_NET_CONFIG
  - SYS_IP_CONFIG (new) for zones
- Can snoop from inside a zone
  - *But can still snoop using the interface in global too!*
- Security Threat
  - Zone can forge ethernet packets

IP Instances – Implementation

- No more global data in IP
- Each instance is tied to **one** zone
  - Cannot have more than one instance per zone
  - Cannot have an instance without a zone
  - Are enabled with zonecfg when creating a zone
    - set ip-type=exclusive
- On OpenSolaris as part of the Crossbow project
- Some network is still Global Zone only:
  - Third party insert below IP, IP QoS, Layer 2 related (eg: link aggregation), CGTP

Zones Networking – IP Instances



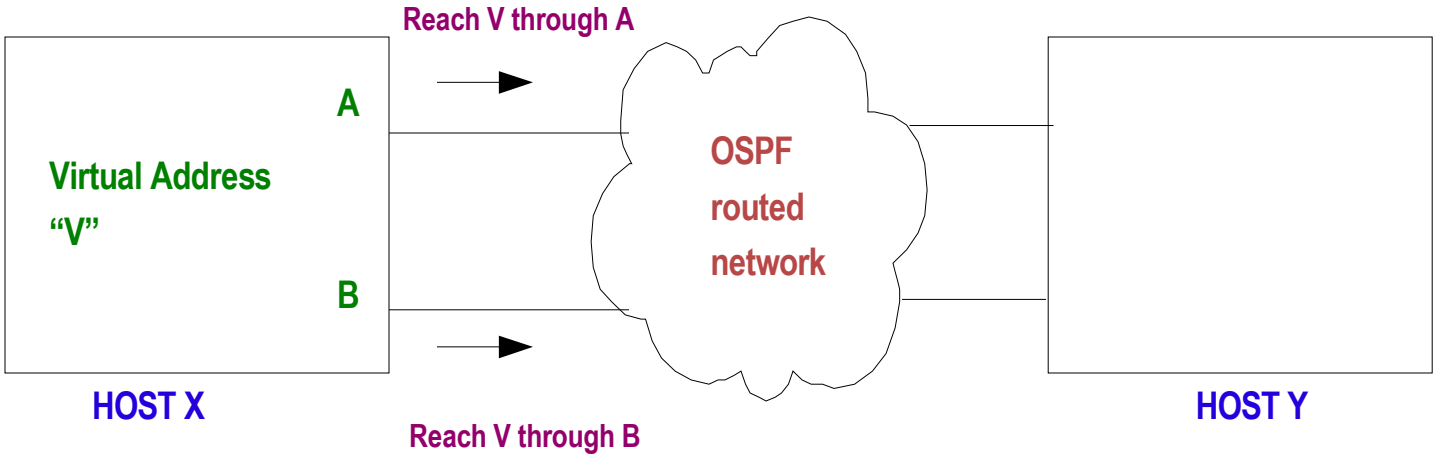
Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

OSPF-MP – Layer 3 Redundancy

- Virtual IP address and interface required
- “ifconfig usesrc” applied to real interfaces we want redundancy for
- OSPF-MP advertises virtual IP address reachable through real interfaces.
  - If a real interface fails, that advertisement is withdrawn

OSPF-MP – Diagram



·Locally generated packets on Host X use "V" as source address

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

MultiData Feature

- Large Segment offload like functionality without added H/W
- Less cycle per instructions
- Improves bulk data transfer performance:
  - Up to ~50% decrease in CPU utilization
  - Improves NFS large writes (server-side) performance
- Significantly improves performance for backups over networking (up to 50% over FireEngine improvements)
- Currently implemented with Sun's 1Gb NIC but will be available for off the shelf NICs with Nemo

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - IPFilter
  - Yosemite
  - Surya
  - SIP
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

Network Layer 7 Cache (NL7C)

- NCA merge with FireEngine
- Create an object cache for web servers
- Next generation NCA using FireEngine TCP/IP stack
- Works with real web servers like Sun One web server, Apache, and Zeus
- Next NL7CS to replace NCAS (turbo change SSL processing)

Solaris Networking – Agenda

- Completed
  - FireEngine
  - SCTP
  - SIP
  - IPFilter
  - Yosemite
  - Surya
  - Updated MIBs
  - Nemo
  - IPv6 Certification
- Upcoming
  - Crossbow
  - IP Instances

IP Duplicate Address Detection- DAD

- Current state of affairs:
  - Newly configured address announced through gratuitous ARP
  - Message logged (at original owner's console) but no action taken
  - Result is network communication chaos
- Solution:
  - DAD checking is done when interface marked UP + RUNNING or address changes

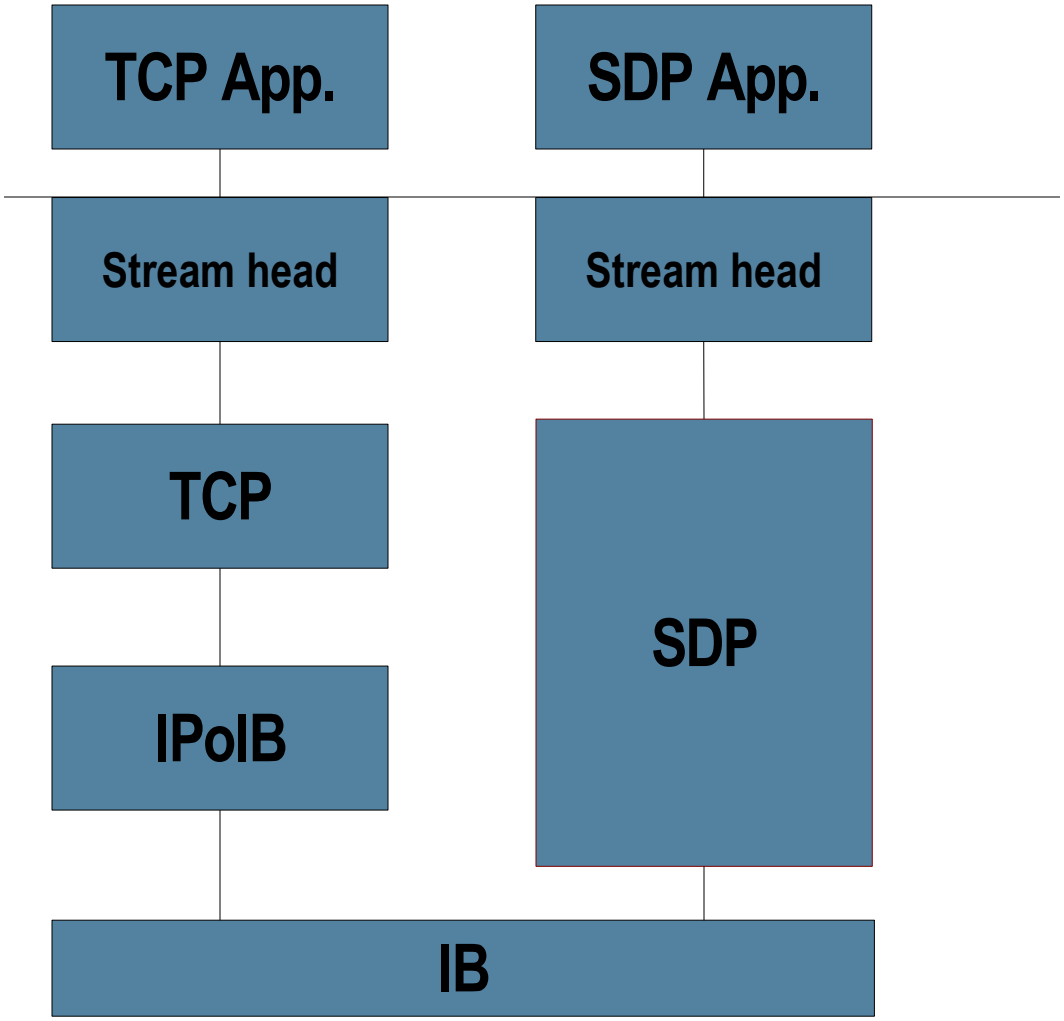
IP Duplicate Address Detection- DAD

- Solution (contd.)
  - Two phases:
    - Validate that the proposed address is not in use
    - Ongoing conflict detection and resolution
  - New flag IFF_DUPLICATE set, interface downed upon failure
  - Routing socket listeners informed only after validation
- RFC 3927 implemented (only the DAD portion, not link local addressing) with some differences. An Internet draft will capture these differences.

Socket Direct Protocol (SDP)

- SDP enables Internet applications to take advantage of the low-latency, high bandwidth performance benefits of RDMA and kernel bypass
- Introduces new SDP_PROTO which the application has to specify
- Supports RDMA semantics but not zero-copy (as of yet)
- Anticipate at least 50% improvement over TCP/IPoIB

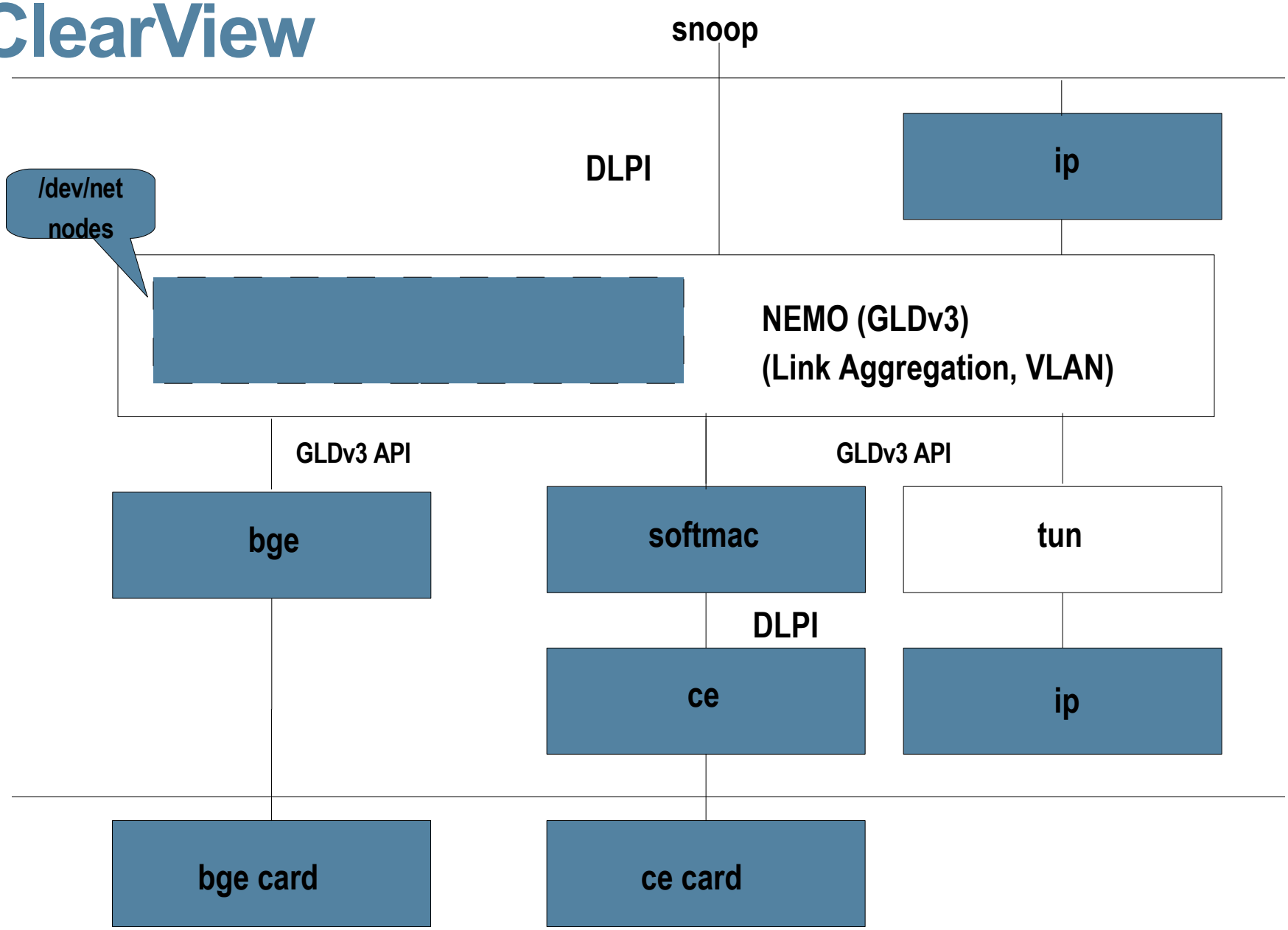
Socket Direct Protocol (SDP)



ClearView

- Nemo (GLDv3) unification
  - Introduces GLDv3 shim driver (softmac)
  - All data-links become GLDv3
    - Can create VLANs/aggregations with any network interface
- Vanity Naming
  - Network interface names can be renamed
  - No longer tied to hardware chipset/port location
    - dladm create-aggr -l bge0 -l bge1backup
- IP Tunnels converted to GLDv3

ClearView



ClearView

- Next generation IPMP
  - IPMP provides high availability and improved network utilization
  - Challenges:
    - Applications needs to be modified
    - DHCP don't work
    - Administration of observability frustrating
- NG IPMP
  - Each IPMP group will have single IP interface
  - ipmpstat: tool to determine health and utilization of IPMP groups

ClearView

- IP traffic unobservable in many cases:
  - Intra-zone, inter-zone, loopback
  - Solution: DLPI observability devices for each IP interface
  - /dev/ipnet/lo0; /dev/ipnet/bge0 etc
  - Just enough DLPI to support observability
  - For zones, /dev/ipnet only that zone's IP interface
  - Traffic to/from addresses hosted on that interface (or forwarded through it) observable
- Nemo generalization
  - Provide MAC-type plugin framework
  - Planned plugins: Ethernet, IP tunnel, WiFi, Infiniband

ClearView

- VLAN observability:
  - VLAN headers, VLAN traffic on data-link not observable
  - Revise semantics, implementation; snoop to display/filter VLAN traffic
- libdlpi.so
  - 9 distinct implementation within source base
  - Library consolidates this into one implementation, shields low level details

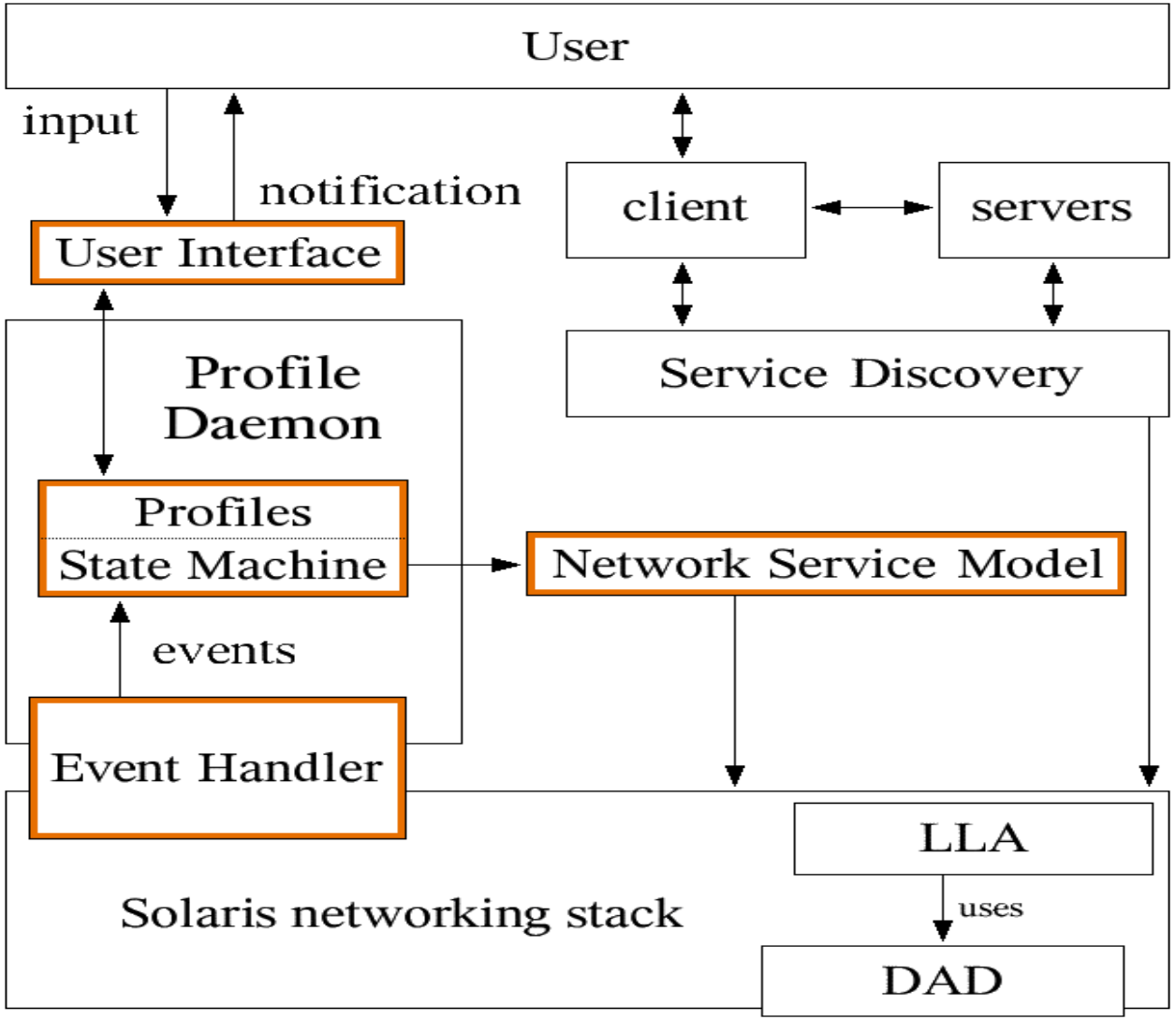
ARP/IP merge

- Motivation:
  - Sustainability: several escalations from ARP and IP having their own caches
  - Code simplification: EOL of ATM driver from Solaris nevada, allows removal of interface used by external modules to talk to ARP
- Solution:
  - Do ARP (putnext-less) processing in IP context
  - Merge split caches into one (Neighbour cache entry)
  - Provide API (ip2mac) to allow external consumers to request ARP to resolve IP address (asynhronous and synchronous mode)

Network AutoMagic (NWAM)

- Introduces:
  - Network Profiles: provide a way to simplify network configuration management
  - Service Discovery: allows network services to be advertised and discovered using the standard DNS packet formats
- Network Profile main components: profile daemon, profiles, event handler, user interface, network service model
- Service Discovery will implement Apples' mDNSResponder

NWAM (Architecture)



Misc

- DHCPv6 client
- IP_NEXTHOP (socket option to specify gateway)Reliable Datagram Service (RDS)
  - Provides rds driver (ported from Linux openIB RDS driver) and socket API
- Enhanced SMF profiles
  - Activation of multiple profiles (based on network events)
- Network Layer 7 Cache
  - Brings NCA into core stack (code simplification!)
- Packet event framework

Misc

- Greyhound
  - In kernel SSL proxy
- IPsec tunnel reform
  - Fixes interopability issues with other implementations (in tunnel mode) by allowing inner IP header selectors
- Multiple MAC address support
  - Ability to set multiple MAC addresses to a network interface (will be used by crossbow VNICs)
- Enhanced Wifi
  - Nemo based dladm configuration
  - Support for more wireless chips and for WPA security

More information...

- http://opensolaris.org
- http://opensolaris.org/os/community/networking/
- Networking stack writeup
  - http://blogs.sun.com/roller/page/sunay
  - http://blogs.sun.com/roller/page/sunay?entry=solaris_networking_the_magic_revealed
  - http://blogs.sun.com/roller/page/sunay?entry=the_solaris_networking_the_magic