



Project Clearview

Peter Memishian

Sebastien Roy

Network Approachability

Sun Microsystems, Inc.

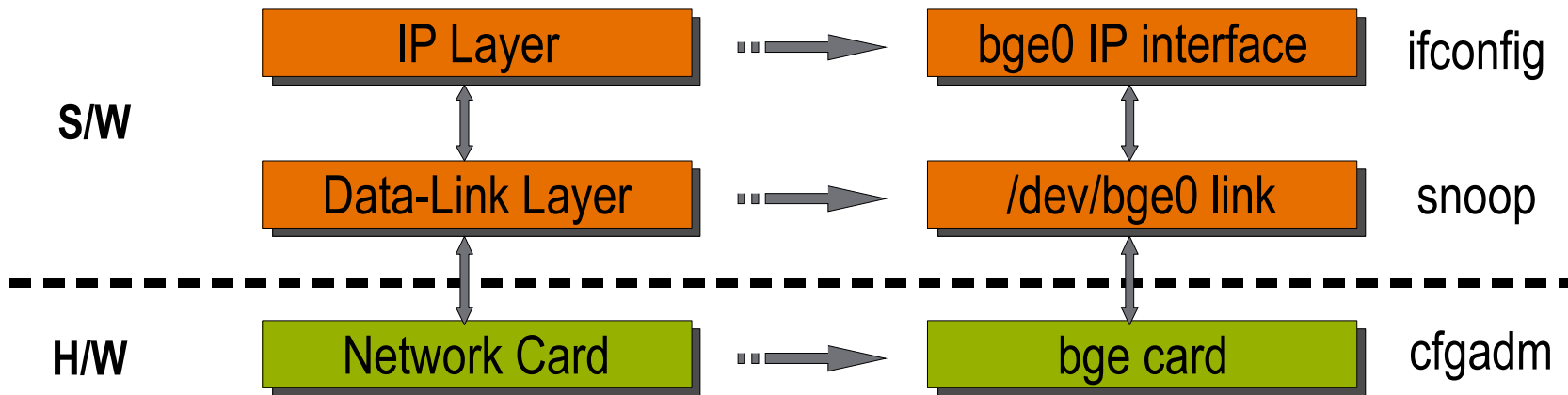
February 2007



Clearview Overview

- Unify, simplify, and enhance the features provided by Solaris networking interfaces
 - > “Network interfaces” as in `ce`, `bge`, `tun`, ...
- Goals:
 - > Unify network interface feature set
 - > Simplify network interface administration
 - > Enhance observability of network interfaces
 - > Increase interoperability between networking features
 - > Improve third-party network application capture

What is a Network Interface?



Network Interfaces: Complaints

- 802.1q VLAN's work with an arbitrary subset of Ethernet networking interfaces.
- 802.3ad Link Aggregation support is even worse:
 - > Some links are aggregated with `d1adm(1M)`
 - > Others are aggregated with the unbundled `nettr(1M)`
 - > Many cannot be aggregated at all!
- Packets cannot be seen on all network interfaces
 - > Cannot see traffic for loopback, tunnels, or IPMP groups
- Network configuration is chipset-dependent
 - > e.g., upgrading `hme` to `bge` means changing `ipfilter` rules

Network Interfaces: More Complaints

- Only some data links are administered with `dladm`
 - > Some – such as IP tunnels – are buried in `ifconfig`
 - > Many cannot be directly administered at all.
- Solaris IPMP – a key part of many high-availability networking deployments – often cannot be used because its odd network interface model breaks:
 - > Dynamic routing daemons
 - > IPsec IKE daemons
 - > IPv6 autoconfiguration
 - > DHCP clients
 - > ... and **countless** third-party applications

Clearview: Network Interface Sanity

- Use VLANs on all Ethernet links
- Use link aggregations across any set of Ethernet links
- Administer all data-links with d1 adm
- Give data-links administratively chosen names
- Observe network traffic on any interface
 - > At the data-link layer, or at the IP layer
- Use IPMP with all the aforementioned technologies
 - > ... **including** countless third-party applications

Clearview Components

- Starring:
 - > Nemo Unification and Vanity Naming (UV)
 - > IP Tunneling Device Driver
 - > Next Generation IPMP
 - > IP Observability Devices
- Supporting Cast:
 - > Nemo Generalization
 - > VLAN Observability
 - > Public DLPI Library

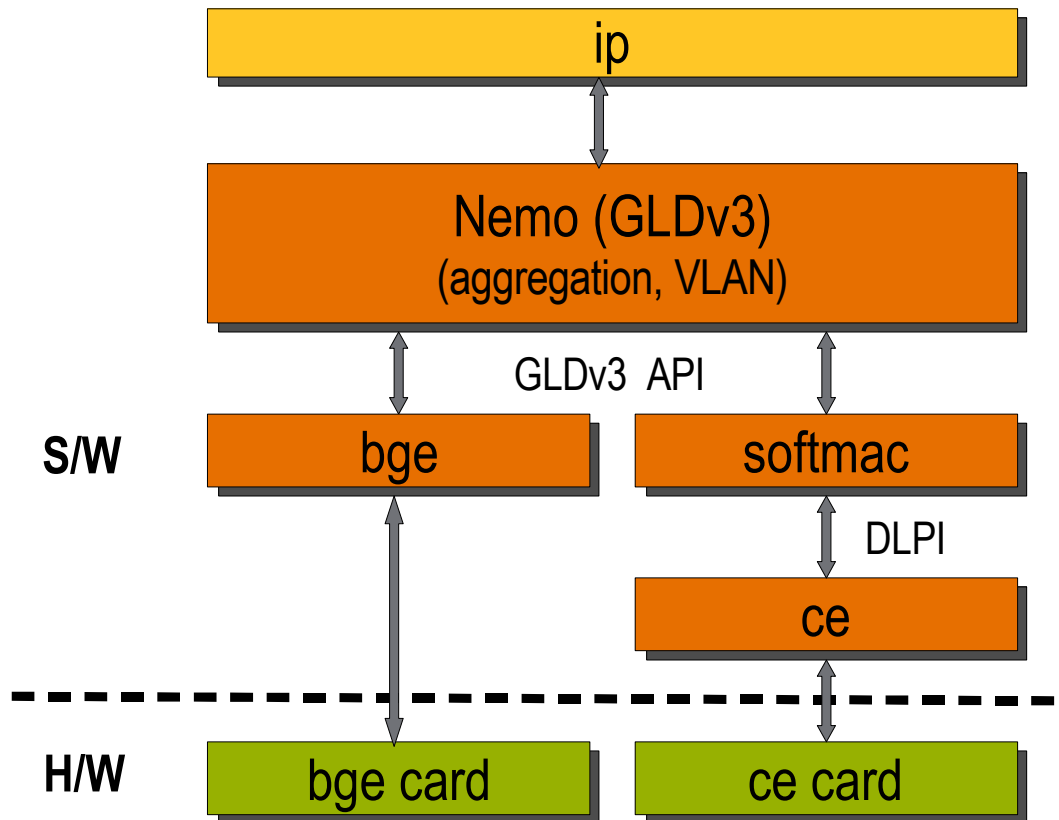
Nemo Unification: Motivation

- Nemo – aka GLDv3 – is our new cutting-edge network device driver framework (S11; S10U1)
 - > Features include VLAN and link aggregation support
- Nemo also introduced `d1adm` (data link admin) to administer GLDv3 links and features
- All new Sun-authored network drivers are GLDv3
 - > But **many** existing network drivers are not
 - > Further, GLDv3 (still) not ready for third-party use
 - > API's still experiencing periodic incompatible changes.

Nemo Unification: Solution

- Solution: introduce a shim driver (`softmac`)
 - > `softmac` is a special GLDv3 network driver
 - > Normal GLDv3 network drivers talk to hardware
 - > Instead, `softmac` talks to an arbitrary DLPI-based network driver
 - > DLPI-based network driver talks to hardware like normal
 - > Requires no changes to underlying driver source or binaries
 - > Requirement: no measurable performance impact for data fastpath
- Brings all data-links under GLDv3
 - > Unified administration under `dladm`
 - > Unified support for VLANs, aggregations
 - > Allows unified support for upcoming GLDv3 features

Nemo Unification: softmac



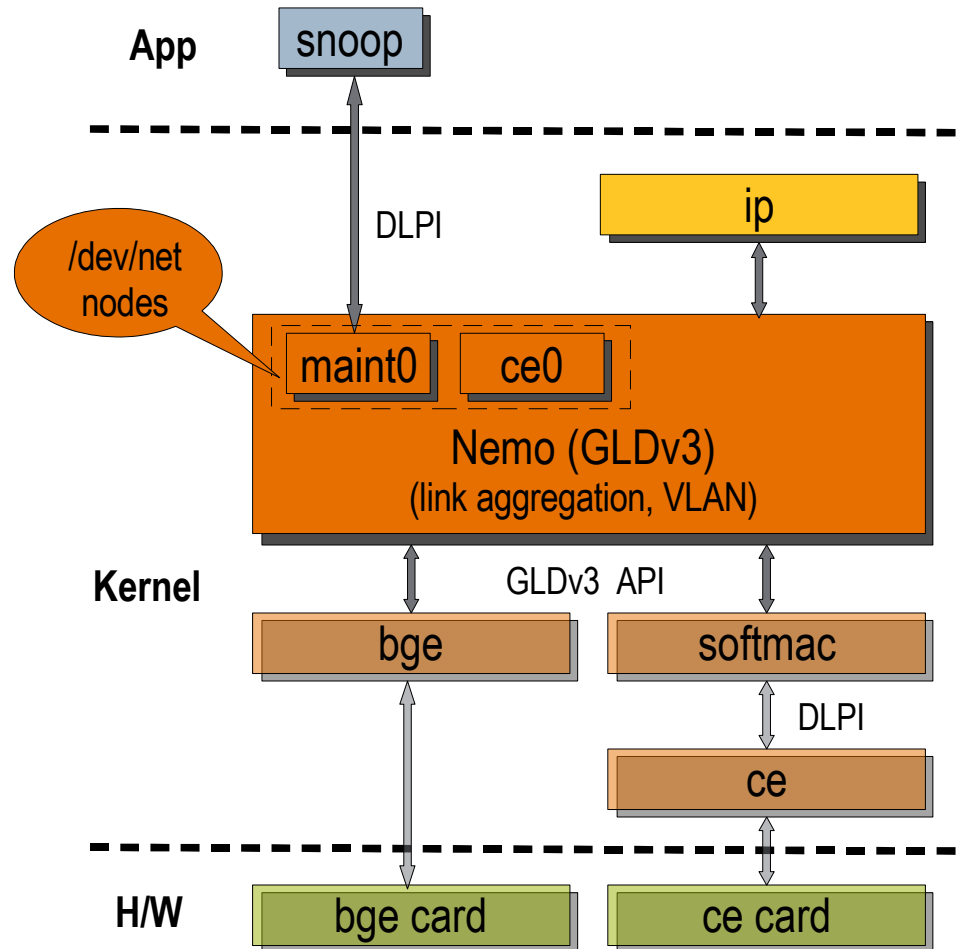
Vanity Naming: Motivation

- Currently, network interface configuration is:
 - > Tied to dozens of hardware chipsets
 - > bge, vge, ce, hme, xgb, ipge, e1000g, ...
 - > Tied to network topology
 - > bge2001, aggr2, ip.tun0
 - > Unportable, even across functionally equivalent systems
 - > Serious issue for zone migration, system cloning, ...
 - > Administratively meaningless
 - > Error-prone, especially on a machine with many interfaces
 - > Impossible to tell system administrative intent of DR operation

Vanity Naming: Solution

- Enhancements to `dladm` to allow links to be named
 - > To rename `bge0` to `maint0`:
 - > `dladm rename-link bge0 maint0`
 - > For compatibility, links default to existing “chipset” names
 - > Further, old DLPI `/dev` names remain fixed (e.g. `/dev/bge`)
 - > New `/dev/net` directory provides vanity name DLPI devices
 - > Every data-link on the system is under `/dev/net`
 - > Can also give VLANs and aggregations vanity names:
 - > `dladm create-aggr -l bge1 -l maint0 a0`
 - > `dladm create-vlan -l a0 -v 2 vlan0`

Vanity Naming: /dev/net nodes



Vanity Naming: Solution

- Vanity Naming is **everywhere**
 - > Nemo Unification allows non-GLDv3 links to be named
 - > Changes to `ifconfig` will plumb DLPI nodes in `/dev/net`
 - > Thus, the vanity name will propagate to the IP layer, and above
- With Vanity Naming, network configuration is:
 - > No longer tied to hardware chipsets
 - > No longer tied to network topology
 - > VLANs and aggregations can have any link name
 - > Portable across functionally equivalent systems
 - > Administratively meaningful

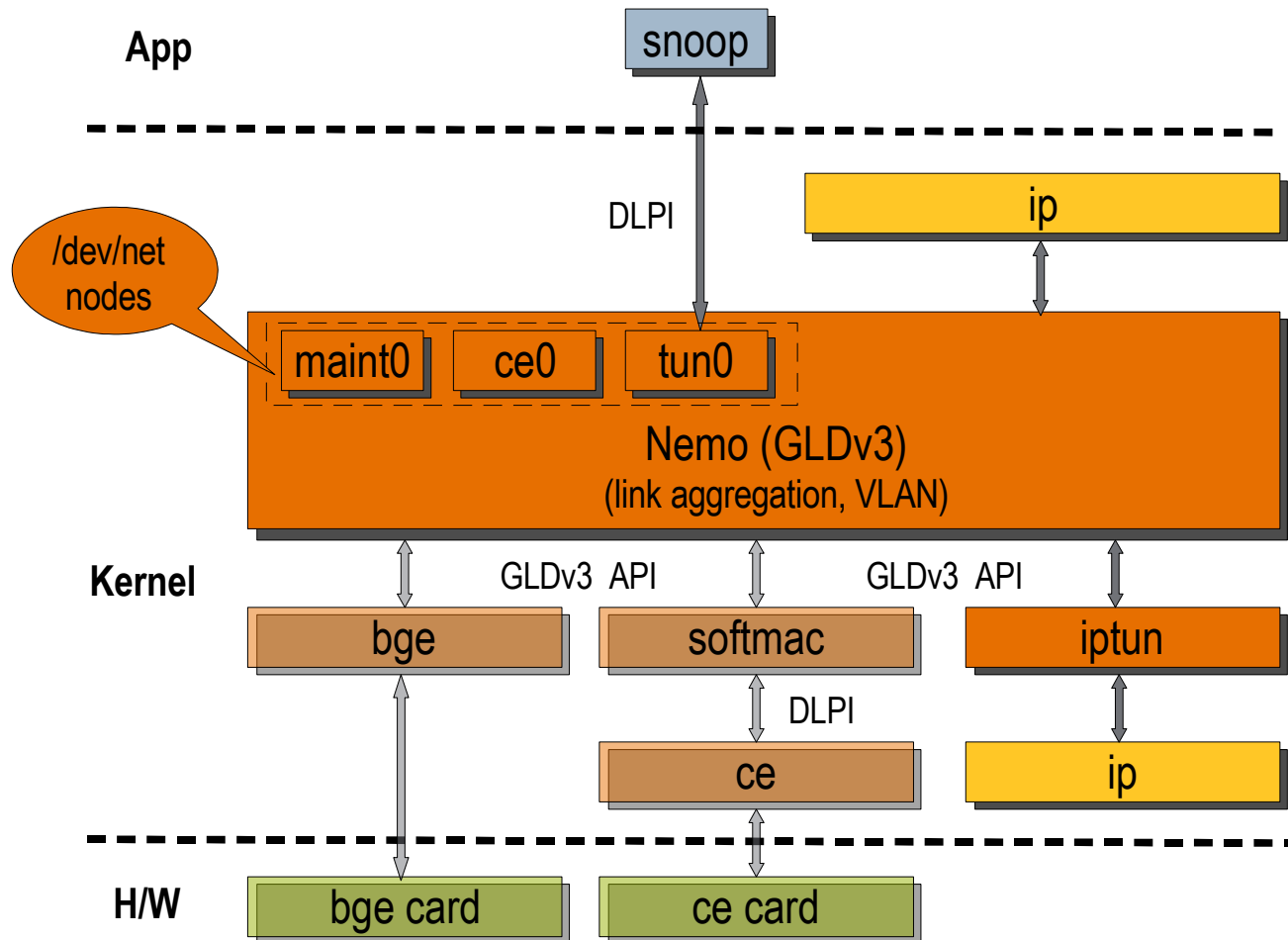
IP Tunneling: Motivation

- Currently, arbitrarily different from other data links:
 - > IP Tunnels do not provide a DLPI /dev node, thus:
 - > IP Tunnel traffic cannot be observed
 - > IP Tunnel traffic cannot be secured via STREAMS firewalls
 - > More generally: DLPI applications cannot be used
 - > IP Tunnel administration is bizarre and limited:
 - > Unlike other link layers, administered through `ifconfig`
 - > Interface names are hard-wired to specific meanings:
 - `ip.tun0`, `ip6.tun0`, `ip.6to4tun0`

IP Tunneling: Solution

- Reimplement IP Tunnels as a GLDv3 network driver
 - > Since all GLDv3 drivers have a DLPI /dev node:
 - > IP Tunnel traffic becomes observable (e.g., through snoop)
 - > DLPI applications and kernel modules can work
 - e.g., STREAMS-based firewalls and IDS's
 - > Since all GLDv3 devices are administered with dladm:
 - > IP Tunnels can be administered like any other link layer
 - For backward compatibility, ifconfig still supported
 - > IP Tunnel interfaces can be named through Vanity Naming

IP Tunneling: GLDv3 driver



IPMP: Solaris IP Network Multipathing

- Intended to provide high-availability and improved network utilization to all IP-based applications:
 - > Without any specialized networking cards
 - > Without any changes to hosts or hardware on network
 - > Without any changes to existing applications
- Like 802.3ad link aggregation, but at the IP-layer:
 - > IP interfaces on the same link are placed into a “group”
 - > Interface health tracked; connections migrated if needed
 - > Focus is on availability rather than performance
- More information: docs.sun.com IP Services Guide

NG IPMP: Motivation

- Widespread IPMP customer use hampered because:
 - > In practice, many applications **do** need to be modified
 - > Does not work with critical networking technologies such as dynamic routing, stateful packet filtering, and DHCP
 - > Frustrating and overwhelming administrative experience
 - > Network observability becomes a nightmare
- Issues caused by two core limitations:
 - > Each IPMP group is modeled as a set of IP interfaces, rather than a single IP interface.
 - > No utility for looking at IPMP status

NG IPMP: Solution

- Make each IPMP group have a single IP interface
 - > Applications interact with the IPMP IP interface
 - > Since it acts like any other IP interface, applications “just work”
 - > Clearview IP Observability component enables network monitors like snoop to interact with the IPMP IP interface
 - > Result: network traffic for a whole IPMP group can be observed
- Tame IPMP administrative experience: `ipmpstat`
 - > Easily determine health of IP interfaces and IPMP groups
 - > Easily determine the configuration and utilization of each IPMP group

IP Observability: Motivation

- Network traffic only observable at data-link layer
 - > Not all IP interfaces have a corresponding data-link
 - > Traffic sent to IP addresses on 100 looped back internally by IP
 - > Not all traffic sent to an IP interface reaches the data-link
 - > Traffic sent to local IP addresses is looped back internally by IP
 - > Not all traffic flows over a predictable data-link
 - > With IPMP, traffic may be load-spread across a set of data links
 - > With IPMP, inbound and outbound traffic for a given connection often use different data-links

IP Observability: Motivation (cont'd)

- Traffic unobservable from inside a zone
 1. Traffic from a zone to another host is unobservable
 - > Traffic flows over data links, but data links do not exist in a zone
 2. Traffic from a zone to another zone is unobservable
 - > Traffic sent to local IP addresses is looped back internally by IP
 3. Traffic inside a zone is unobservable
 - > Traffic sent to IP addresses on lo0 looped back internally by IP
- Traffic for **2** and **3** unobservable from global zone
- IPMP problems painful to diagnose
- Loopback protocol issues painful to diagnose

IP Observability: Solution

- “DLPI” observability devices for each IP interface
 - > Reside in new `/dev/ipnet` directory
 - e.g., `/dev/ipnet/lo0`; `/dev/ipnet/bge0`
 - > Implement just enough DLPI to provide observability
 - NB: Not general DLPI devices – nor can they be!
 - > Provide traffic to/from addresses hosted on that interface, or forwarded through it.
 - > For zones, `/dev/ipnet` only has that zone's IP interfaces
 - Further, only traffic to/from that zone's IP addresses visible
 - > For compatibility with Linux and *BSD: `/dev/lo0`

Supporting Cast: Nemo Generalization

- Motivation: GLDv3 is currently Ethernet-specific
 - > Blocks Nemo Unification and IP Tunnelling Device Driver
- Solution: Introduce MAC-Type plugin framework
 - > MAC-Type plugin framework defines a set of callbacks
 - > Support for each MAC Type provided via a plugin implementing the defined set of callbacks
 - > GLDv3 framework invokes callbacks to perform MAC-Type specific operations
 - > Planned plugins: Ethernet, IP Tunnels, WiFi, InfiniBand
 - > Also has broader aims to make GLDv3 extensible

Supporting Cast: VLAN Observability

- Motivation: VLAN traffic cannot be properly observed
 - > VLAN headers cannot be observed at all
 - > VLAN traffic not observable on data link it's running atop
 - > e.g., `snoop -d bge0` will not show VLAN traffic atop `bge0`
 - > Blocks development of Nemo Unification (`softmac`)
 - > VLAN observability a hot issue with some big customers
- Solution: Revise semantics and implementation
 - > Tedious because the implementation is not centralized
 - > Also: enhance `snoop` to display and filter VLAN traffic

Supporting Cast: Public DLPI Library

- Motivation: Apps have low-level DLPI knowledge
 - > Impacts Vanity Naming and IP Observability
 - > Some applications assume all DLPI nodes are in /dev
 - > More broadly: code maintenance nightmare
 - > 9 distinct implementations of DLPI routines in ON alone
 - > 3rd parties undoubtedly have more in their applications
- Solution: Introduce documented DLPI API
 - > Shields low-level DLPI details from applications
 - > Centralizes low-level DLPI logic
 - > Can now be maintained – and evolved.

Status and Schedule

- Initial designs complete; under active development
 - > EA bits now available via opensolaris.org
- Preliminary Solaris 11 integration dates:
 1. Nemo Generalization: Integrated (onnv_44)
 2. VLAN Observability: Integrated (onnv_50)
 3. Public DLPI Library: Integrated (onnv_59)
 4. IP Observability Devices: Spring 2007 (onnv_64)
 5. Vanity Naming + Nemo Unification: Spring 2007 (onnv_65)
 6. IP Tunneling Device Driver: Spring 2007 (onnv_66)
 7. NG IPMP: Summer 2007 (onnv_72)
- 1-3, 4 are likely additions to future S10 Updates

Clearview: More Information

- Internal Wiki
 - > <http://clearview.east/>
 - > Detailed schedule; mail archives; meeting minutes
- OpenSolaris Clearview Project
 - > <http://opensolaris.org/os/projects/clearview>
 - > Overview; design documents; links to design discussion
- Mailing Lists
 - > I-team: clearview-iteam@sun.com
 - > External: clearview-discuss@opensolaris.org



Project Clearview

Peter Memishian

Sebastien Roy

clearview-iteam@sun.com

