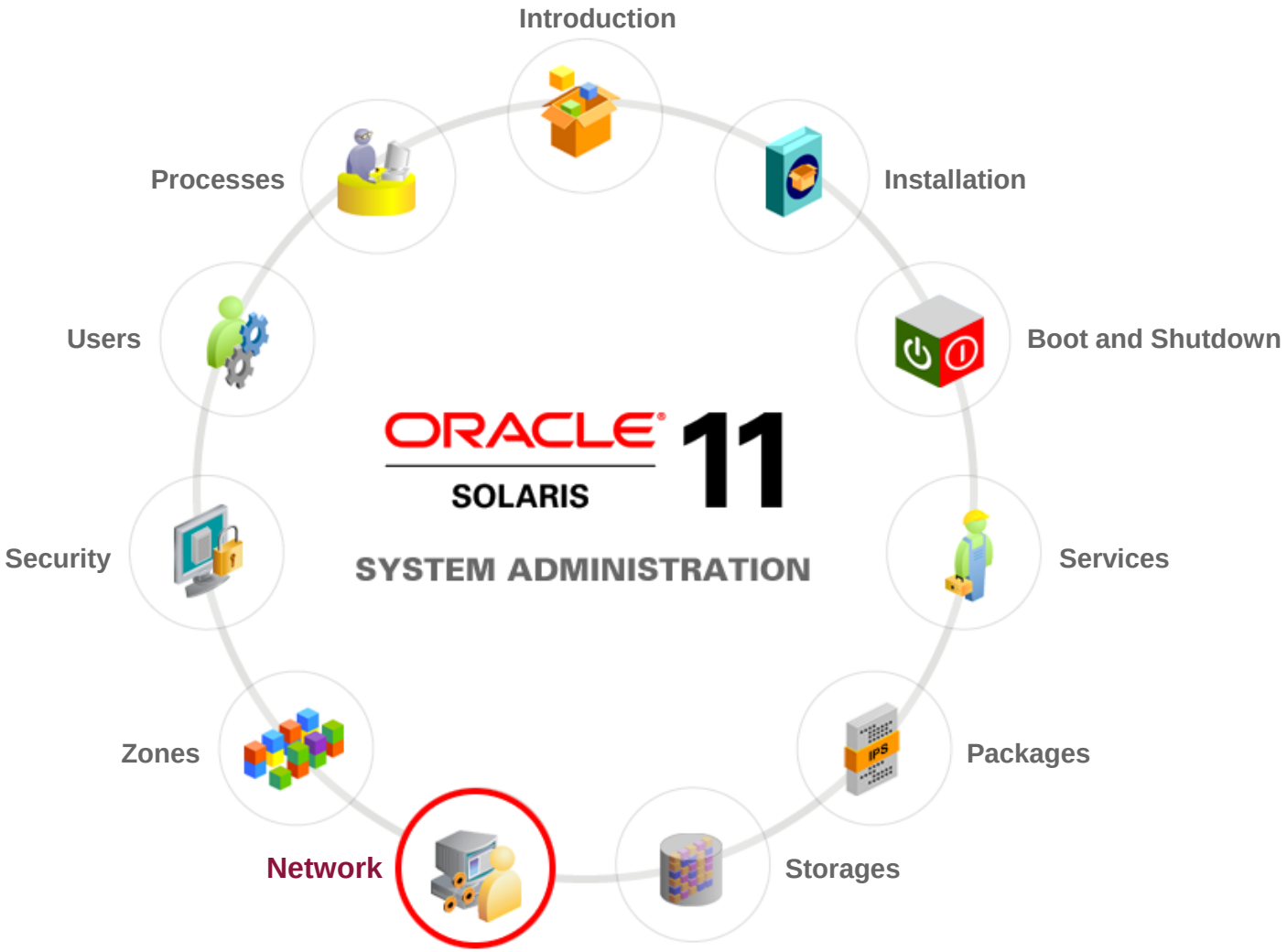
7

Administering the Network

Workflow Orientation



Objectives

After completing this lesson, you should be able to:

- Explain some of the basic networking concepts
- Administer a datalink configuration
- Administer a network interface
- Administer a profile-based network configuration
- Configure a virtual network
- Verify network operations
- Manage resources on the network

Agenda

- **Reviewing Networking Fundamentals**
- Administering a Datalink Configuration
- Administering a Network Interface
- Administering a Profile-Based Network Configuration
- Configuring a Virtual Network
- Verifying Network Operations
- Managing Resources on the Virtual Network

Importance of Network Administration

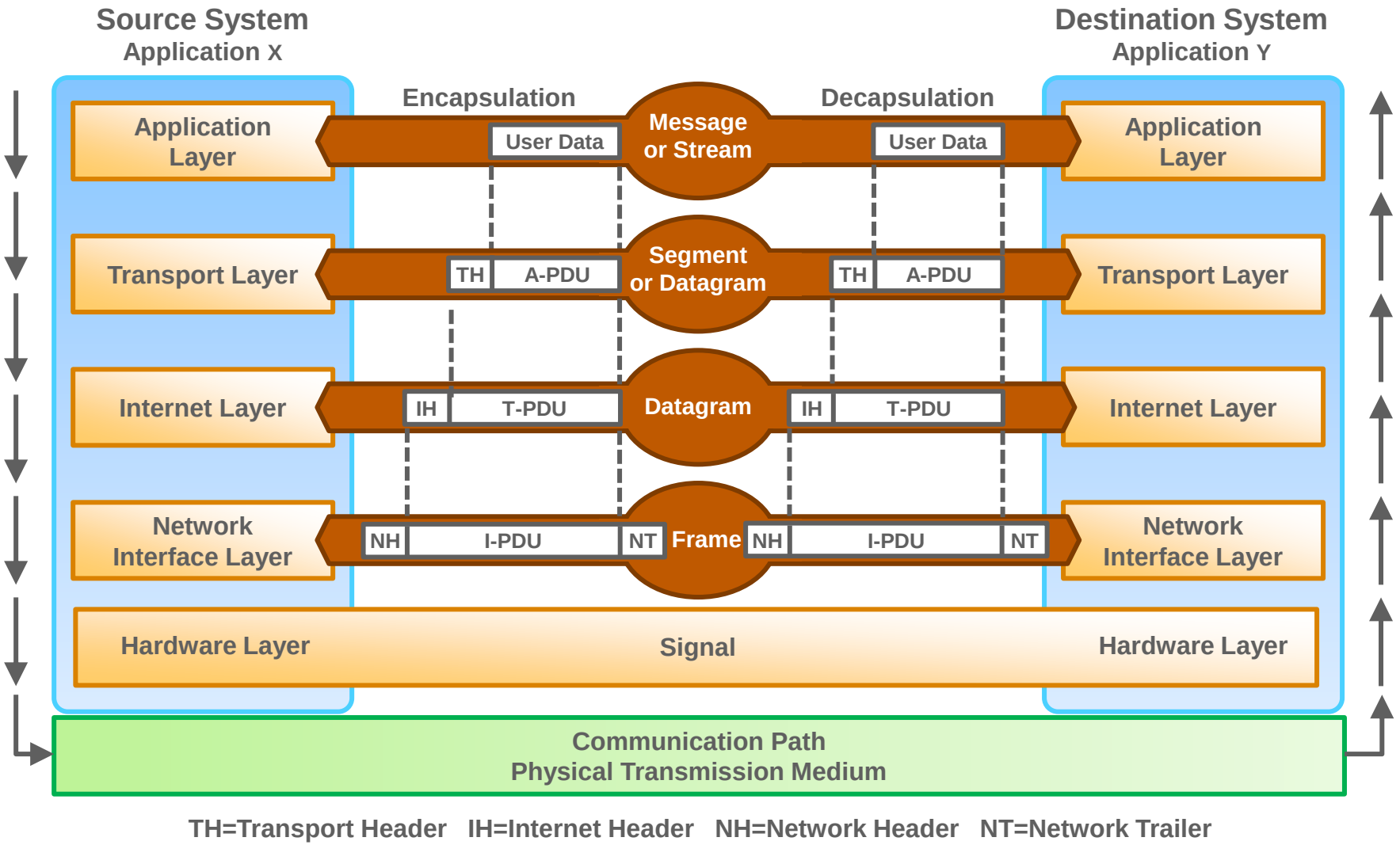
It is important to administer the network in the Oracle Solaris 11 OS to address the following requirements:

- IP addressing scheme
- Network interfaces
- Datalinks
- Network configuration profiles
- Virtual networks
- Network resources

TCP/IP Protocol Architecture Model

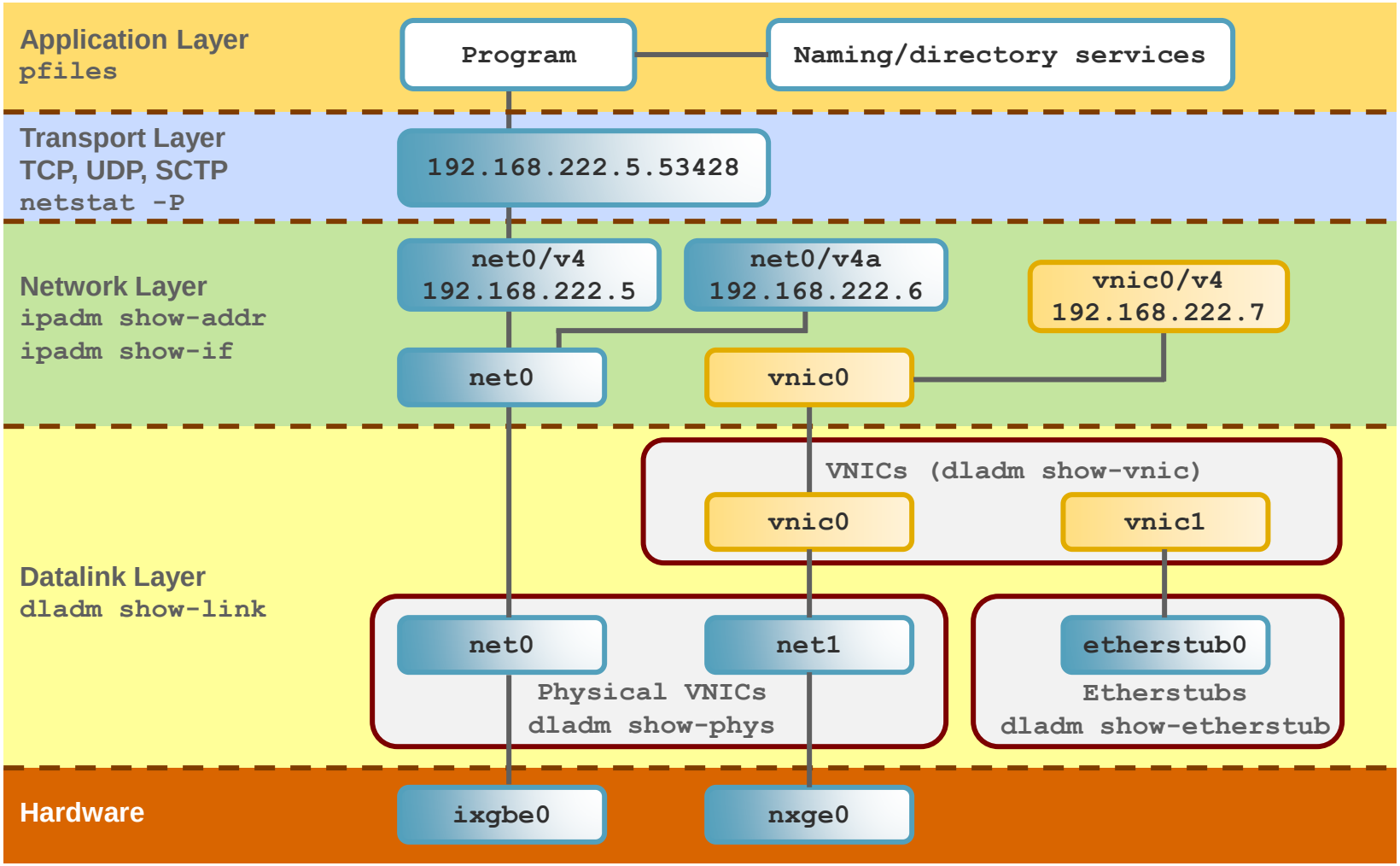
OSI Ref. Layer No.	OSI Layer Equivalent	TCP/IP Layer	TCP/IP Protocol Examples
5, 6, 7	Application (7) Presentation (6) Session (5)	Application	`telnet`, `ftp`, `rlogin`, DNS, LDAP, and NFS
4	Transport	Transport	TCP
3	Network	Internet	IPv4, IPv6
2	Datalink (2)	Datalink	IEEE 802.2. Ethernet (IEEE 802.3)
1	Physical	Physical Network	

How TCP/IP Handles Data Communications



TH=Transport Header IH=Internet Header NH=Network Header NT=Network Trailer

Oracle Solaris 11 Networking Stack



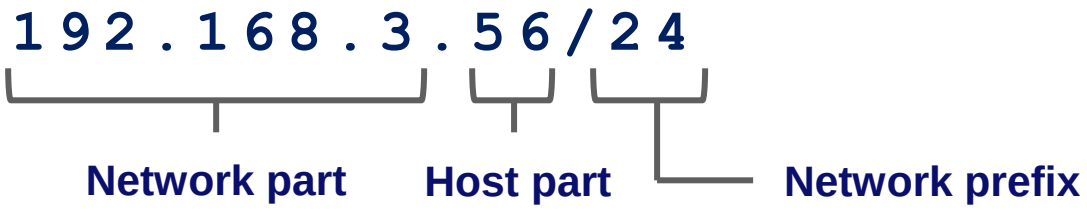
Configuring a Host for TCP/IP

Network configuration checklist:

- ☑ IP addresses
- ☑ Netmask
- ☑ Domain name
- ☑ Name service
- ☑ **Default router**

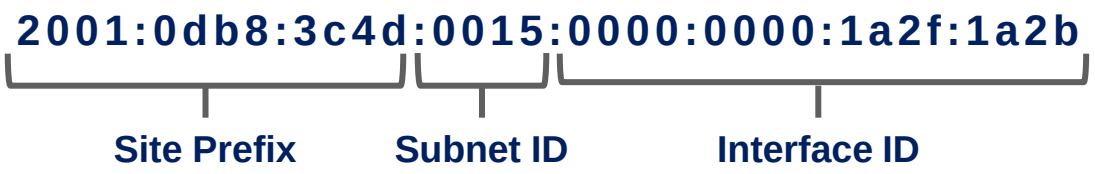
IPv4 Addressing

- The IPv4 address is:
  - A 32-bit number that uniquely identifies a network interface on a system
  - Written in decimal digits
  - Divided into four 8-bit fields that are separated by periods
- The component parts of an IPv4 address include:
  - Network part
  - Host part
  - Network prefix



IPv6 Addressing

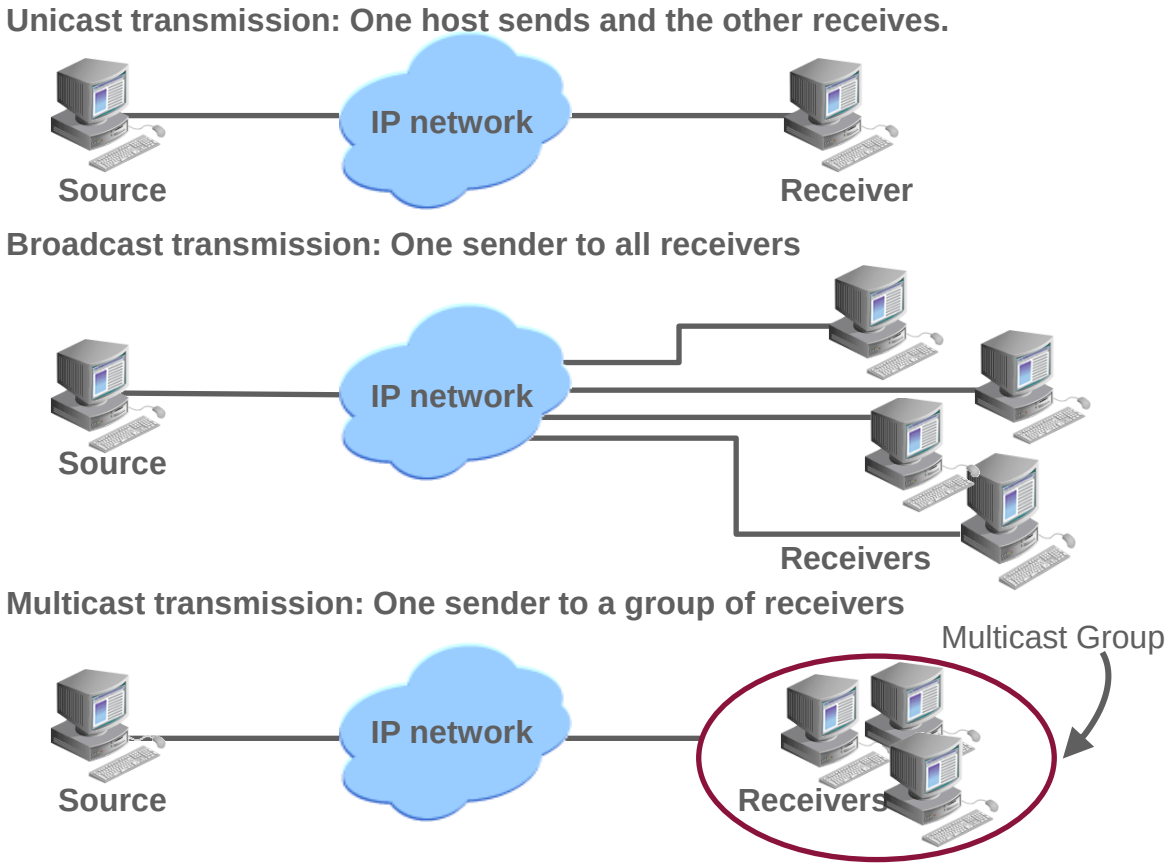
- Was developed to address:
  - IPv4 shortage
  - Manual address configuration
- Uses 128-bit addressing
  - Divided into eight 16-bit fields, with each field bounded by a colon
  - Written in hexadecimal numbers
- Includes component parts such as:
  - Site prefix
  - Subnet ID
  - Interface ID



Unicast, Multicast, and Broadcast Addressing

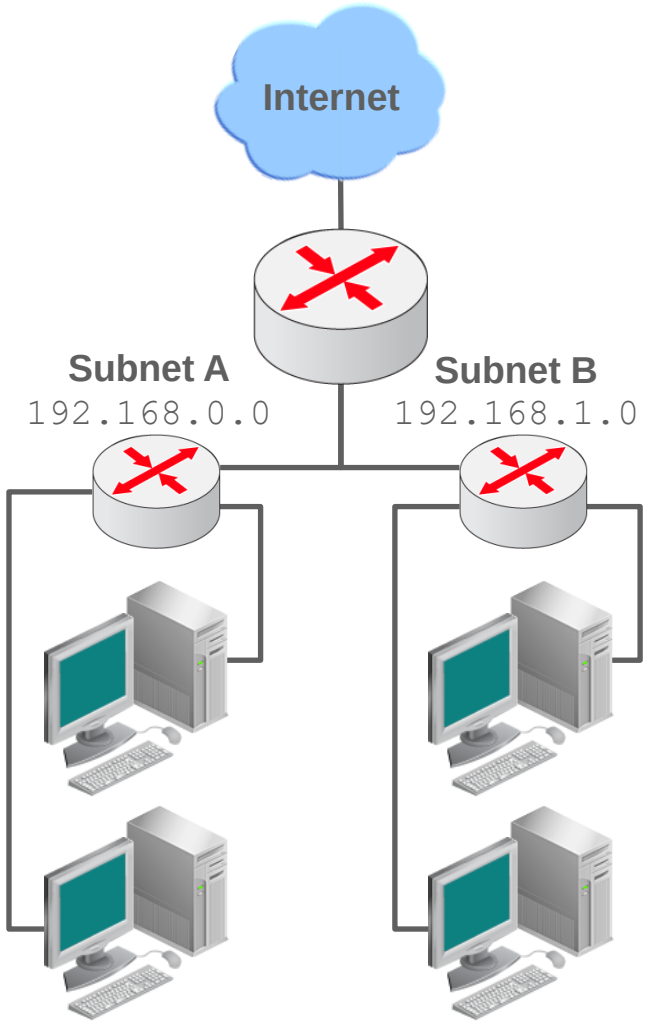
For each type of data transmission, there is an associated IP addressing type:

- Unicast
- Broadcast
- Multicast



Subnets, Netmasks, and Subnet Masks

- Subnets:
  - Allow allocation of the host address space to network addresses
  - Are created by using a netmask
- Netmasks determine:
  - How many and which bits in the host address space represent the subnet number
  - How many and which bits represent the host number
- Subnet masks determine which bits in the host address bytes are applied to the subnet and host addresses.



Network Configuration Modes

The network configuration modes refer to the ability of the system to automatically adjust to changes in the current network environment and not to whether static or fixed IP addresses can be configured in these modes.

The following network configuration modes are supported in Oracle Solaris 11:

- Fixed
- Reactive

Oracle Solaris 11 Network Administration Commands

Command	Description
`dladm`	Used to administer datalinks. It helps in managing physical interfaces (Ethernet, wireless, and InfiniBand), virtual networking features (Etherstubs, VNICs, and IP tunnels), switch features (link aggregations, VLANs, VXLANs, and bridging technologies), and device characteristics (speed, duplexing, priority, and feature negotiation).
`ipadm`	Used to administer IP interfaces and IP addresses
`netcfg`	Used to manage various types of profiles, for example, NCPs and location profiles
`netadm`	Used to enable and disable profiles and display information about profiles and their states

Administering the Network

As part of network administration, you will now learn how to:

- Administer datalink configuration
- Administer the network interface
- Administer profile-based network configuration
- Configure a virtual network
- Verify network operation
- Manage resources on the virtual network

Quiz

Which layer of the TCP/IP protocol stack is responsible for accepting and delivering packets for the network?

a. Datalink
b. Transport
c. Internet
d. Application

Quiz

Which layer of the TCP/IP protocol stack is responsible for accepting and delivering packets for the network?

a. Datalink
b. Transport
c. Internet
d. Application

Quiz

The TCP/IP protocol supports only IPv4 addressing.

a. True
b. False

Quiz

The TCP/IP protocol supports only IPv4 addressing.

a. True

b. False

Quiz

This is an example of an IPv4 address: 192.168.3.56/24

a. True
b. False

Quiz

This is an example of an IPv4 address: 192.168.3.56/24

a. True
b. False

Agenda

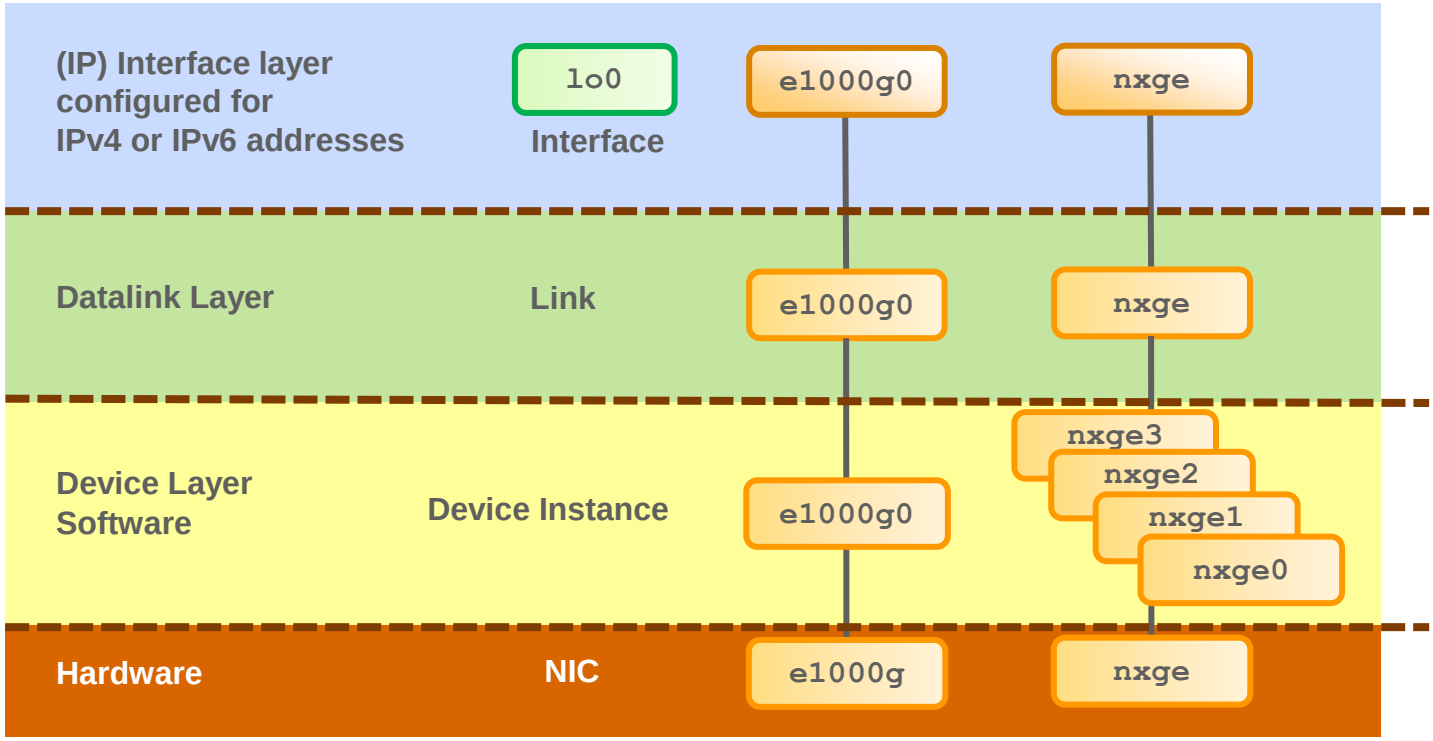
- Reviewing Networking Fundamentals
- **Administering a Datalink Configuration**
- Administering a Network Interface
- Administering a Profile-Based Network Configuration
- Configuring a Virtual Network
- Verifying Network Operations
- Managing Resources on the Virtual Network

Datalink Configuration in Oracle Solaris11

- Administrators create IP interfaces on top of datalinks.
- Each datalink represents a link object in the second layer of the Open Systems Interconnection (OSI) model.
- Datalinks can represent many different Layer 2 entities such as physical network devices (termed physical links), aggregations of physical datalinks, virtual network interface cards (VNICs), and so on.

Determining Datalink Availability

- Determining the physical links that are available
- Determining the datalinks that are available
- Verifying that the network service is running



Determining the Physical Links That Are Available

To display information about the physical attributes of datalinks, use `dladm show-phys`.

```
# dladm show-phys
LINK         MEDIA        STATE      SPEED  DUPLEX    DEVICE
net1         Ethernet     up         1000   full      e1000g1
net2         Ethernet     up         1000   full      e1000g2
net0         Ethernet     up         1000   full      e1000g0
net3         Ethernet     down       1000   unknown   e1000g3
```

Determining the Datalinks That Are Available

To check the status of the datalinks, use `dladm show-link`.

```
# dladm show-link
LINK        CLASS     MTU    STATE    OVER
net0        phys      1500   up       --
net1        phys      1500   up       --
net2        phys      1500   up       --
net3        phys      1500   down     --
```

Verifying That the Network Service Is Running

To verify that the network service is running, use `svcs network/physical`.

```
# svcs network/physical
online          3:33:46 svc:/network/physical:upgrade
online          3:33:53 svc:/network/physical:default
```

Quiz

Which utility is used to create virtual switches and VNICs?

a. `lnkadm`

b. `dladm`

c. `vniccfg`

d. `dlcfg`

Quiz

Which utility is used to create virtual switches and VNICs?

a. `lnkadm`

b. `dladm`

c. `vniccfg`

d. `dlcfg`

Agenda

- Reviewing Networking Fundamentals
- Administering a Datalink Configuration
- **Administering a Network Interface**
- Administering a Profile-Based Network Configuration
- Configuring a Virtual Network
- Verifying Network Operations
- Managing Resources on the Virtual Network

Administering a Network Interface

- Displaying network interface configuration information
- Displaying network interface IP address information
- Configuring a physical network interface manually
- Taking down a network interface
- Bringing up a network interface
- Deleting a physical network interface manually

Displaying Network Interface Configuration Information

To display information about the current network interface configuration, use `ipadm show-if`.

```
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE     OVER
lo0        loopback ok       yes        --
net0       ip       ok       yes        --
net1       ip       ok       yes        --
net2       ip       ok       yes        --
```

Displaying Network Interface IP Address Information

To display network interface IP address information, use `ipadm show-addr`.

```
# ipadm show-addr
ADDROBJ       TYPE     STATE        ADDR
lo0/v4        static   ok           127.0.0.1/8
net0/v4       static   ok           192.168.1.200/24
net1/v4       static   ok           192.168.1.201/24
net2/v4       static   ok           192.168.1.202/24
lo0/v6        static   ok           ::1/128
net0/v6       addrconf ok           fe80::a00:27ff:fe68:6f2d/10
```

Configuring a Physical Network Interface Manually

1. Check the current status of the `network/physical:default` service by using `svcs network/physical`. If the service is not up and running, enable it by using `svcadm enable network/physical:default`.
2. Create a network interface by using `ipadm create-ip` *`interface`*.
3. Specify the IP address by using `ipadm create-addr -T static -a` *`addrobj`*.
4. Verify the network interface configuration by using `ipadm show-if`.
5. Verify the IP address information by using `ipadm show-addr`.

Configuring a Physical Network Interface Manually: Example

```
# svcs network/physical
STATE          STIME    FMRI
online         9:34:40 svc:/network/physical:default
# ipadm create-ip net0
# ipadm create-addr -T static -a 192.168.1.200/24 net0/v4add1
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
net0       ip       ok       yes    --
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4add1       static   ok           192.168.1.200/24
lo0/v6            static   ok           ::1/128
```

Taking Down an Address

To take an address down, use `ipadm down-addr` *`addrobj`*.

```
# ipadm down-addr net3/v4
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           192.168.1.200/24
net1/v4           static   ok           192.168.1.201/24
net2/v4           static   ok           192.168.1.202/24
net3/v4           static   down         192.168.1.203/24
lo0/v6            static   ok           ::1/128
net0/v6           addrconf disabled     ::
```

Bringing Up an Address

To bring up an address, use `ipadm up-addr` *`addrobj`*.

```
# ipadm up-addr net3/v4
# ipadm show-addr
ADDROBJ           TYPE     STATE       ADDR
lo0/v4            static   ok          127.0.0.1/8
net0/v4           static   ok          192.168.1.200/24
net1/v4           static   ok          192.168.1.201/24
net2/v4           static   ok          192.168.1.202/24
net3/v4           static   ok          192.168.1.203/24
lo0/v6            static   ok          ::1/128
net0/v6           addrconf disabled    fe80::a00:27ff:fe68:6f2d/10
```

Deleting a Physical Network Interface Manually

1. Delete the IP address by using `ipadm delete-addr` *`addrobj`*.
2. Delete the network interface by using `ipadm delete-ip` *`interface`*.
3. Verify that the network interface is deleted by using `ipadm show-if`.
4. Verify that the IP address information is deleted by using `ipadm show-addr`.

Deleting a Physical Network Interface Manually: Example

```
# ipadm delete-addr 192.168.1.200/24 net0/v4add1
# ipadm delete-ip net0
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
lo0/v6            static   ok           ::1/128
```

Summary of `ipadm` Commands

Network Interface Task	`ipadm` Command
Display network interface information.	`ipadm show-if`
Display IP address assignments to network interfaces.	`ipadm show-addr`
Create a network interface.	`ipadm create-ip` *interface*
Assign a static IP address to a network interface.	`ipadm create-addr -T` *address-type* `-a` *address/prefixlen addrobj*
Take down an address.	`ipadm down-addr` *addrobj*
Bring up an address.	`ipadm up-addr` *addrobj*
Delete an IP address assigned to a network interface.	`ipadm delete-addr` *addrobj*
Delete a network interface.	`ipadm delete-ip` *interface*

Agenda

- Reviewing Networking Fundamentals
- Administering a Datalink Configuration
- Administering a Network Interface
- **Administering a Profile-Based Network Configuration**
- Configuring a Virtual Network
- Verifying Network Operations
- Managing Resources on the Virtual Network

Profile-Based Network Configuration

- Provides a predetermined set of system-defined profiles
- Provides capabilities for creating various types of user-defined profiles
- Provides the following profile types:
  - Network Configuration Profiles (NCPs)
  - Network Configuration Units (NCUs)
  - Location profiles
  - External Network Modifiers (ENMs)
  - Known WLANs

Reactive Network Configuration Mode

- A reactive network configuration automatically configures Ethernet and Wi-Fi connections.
- The primary focus of a reactive network configuration is mobility.
- A reactive network configuration automatically manages network configuration by storing information in the form of *profiles* in the system.
- You use the `netcfg` and `netadm` commands to create and customize new profiles.

How Reactive Network Profiles Work

- The system provides the `Automatic` NCP and the location profile as the default reactive profiles.
- The automatic or reactive network configuration is triggered by an event or activity.
- The profiles perform a basic configuration of your wired or wireless network automatically, without any user interaction.

Interaction of Reactive Networking with Other Oracle Solaris Networking Technologies

- Network virtualization
  - Virtual machines: Oracle VM Server for SPARC (formerly Logical Domains) and Oracle VM VirtualBox
  - Oracle Solaris zones and stack instances
- Dynamic Reconfiguration (DR) and NCPs
- Fixed network configuration mode commands

`netcfg` Command

`netcfg` Subcommand	**Description**
`create`	Create an in-memory profile of the specified type and name.
`select` ***`object-type`***	Select the profiles that are available at the current scope level and move into that object's scope.
`walkprop`	Walk each property associated with the current profile. For each property, the name and current value are displayed, and a prompt is given to allow the user to change the current value.
`set` ***`prop-name=value1`***	Set the current (in-memory) value of the specified property. If the process is performed in non-interactive mode, the change is also committed to persistent storage.
`list`	List all profiles, property-value pairs, and resources that exist at the current or specified scope.

`netcfg` Command

`netcfg` Subcommand	**Description**
`verify`	Verify that the current in-memory object has a valid configuration.
`commit`	Commit the current profile to persistent storage.
`end`	End the current profile specification, and move to the next higher scope.
`Exit`	Exit the `netcfg` session. The current profile is verified and committed before ending.
`Destroy`	Remove the specified profile from memory and persistent storage.

`netadm` Command

`netadm` Subcommand	**Description**
`enable`	Enable the specified profile. If the profile name is not unique, the profile type must be specified to identify the profile that is to be enabled.
`disable`	Disable the specified profile. If the profile name is not unique, the profile type must be specified to identify the profile that is to be disabled.
`list`	List all available profiles and their current state. If a profile is specified by name, list the current state of only that profile.
`show-events`	Listen for a stream of events from the NWAM daemon and display them.
`scan-wifi`	Initiate a wireless scan on link *linkname*.
`select-wifi`	Select a wireless network to connect to, from the scan results on link *linkname*. You may be prompted for selection, WiFi key, and so forth, if necessary.
`help`	Display a usage message with short descriptions for each subcommand.

SMF Network Services

In Oracle Solaris 11, network configuration is implemented by multiple SMF services as follows:

Service	Description
svc:/network/loopback:default	Creates the IPv4 and IPv6 loopback interfaces
svc:/network/netcfg:default	Manages the network configuration repository, with its primary function being to start the `netcfgd` daemon. This service is a prerequisite for the `svc:/network/physical:default` service.
svc:/network/physical:default	Brings up links and plumbs IP interfaces. This service starts the network management daemon, `nwamd`.
svc:/network/location:default	Enables the location profile that is selected by the `nwamd` daemon. This service is dependent on the `svc:/network/physical:default` service.

Configuring a Reactive Network

This section covers the following topics:

- Configuring a network configuration profile
- Creating a location profile
- Listing a location profile
- Modifying profiles
- Listing reactive network profiles
- Enabling and disabling profiles
- Displaying profile states
- Querying profile information
- Creating a backup of a profile
- Removing reactive network profiles

Creating a Network Configuration Profile

To create an NCP, use the `netcfg` utility.

```
# netcfg
netcfg> create ncp my_profile
netcfg:ncp:my_profile> create ncu phys net1
Created ncu 'net1'. Walking properties …
activation-mode (manual) [manual|prioritized]> manual
mac-address> <ENTER>
autopush> <ENTER>
mtu> <ENTER>
netcfg:ncp:my_profile:ncu:net1> list
ncu:net1
 type                 link
class                 phys
parent               ”my_profile”
activation-mode      manual
enabled              true
netcfg:ncp:my_profile:ncu:net1> end
Committed changes
netcfg:ncp:my_profile> list
ncp:my_profile
        management-type      reactive
NCUs:
       phys  net1
netcfg:ncp:my_profile> exit
```

Creating a Location Profile

Use the `netcfg` utility as follows:

```
# netcfg
netcfg> create loc office
Created loc 'office'.  Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
conditions> "system-domain is example.com"
nameservices (dns) [dns|files|nis|ldap]> dns
nameservices-config-file ("/etc/nsswitch.dns")> <ENTER>
dns-nameservice-configsrc (dhcp) [manual|dhcp]> manual
dns-nameservice-domain> "example.com"
dns-nameservice-servers> "192.168.1.200"
dns-nameservice-search> <ENTER>
dns-nameservice-sortlist> <ENTER>
dns-nameservice-options> <ENTER>
nfsv4-domain> <ENTER>
ipfilter-config-file> <ENTER>
ipfilter-v6-config-file> <ENTER>
ipnat-config-file> <ENTER>
ippool-config-file> <ENTER>
ike-config-file> <ENTER>
ipsecpolicy-config-file> <ENTER>
netcfg:loc:office> list
```

Listing a Location Profile

```
netcfg:loc:office> list
loc:office
          activation-mode                 conditional-all
          conditions                      "system-domain is example.com”
          enabled                         false
          nameservices                    dns
          nameservices-config-file        "/etc/nsswitch.dns”
          dns-nameservice-configsrc       manual
          dns-nameservice-domain          “example.com”
          dns-nameservice-servers         "192.168.1.200”
netcfg:loc:office> verify
All properties verified
netcfg:loc:office> commit
Committed changes
netcfg:loc:office> end
netcfg> exit
```

Modifying Profiles

```
# netcfg
netcfg> select ncp my_profile
netcfg:ncp:my_profile> select ncu net1
netcfg:ncp:my_profile:ncu:net1> list
ncu:net1
              type                    link
              class                   phys
              parent                  "my_profile"
              activation-mode         manual
              enabled                 true
netcfg:ncp:my_profile:ncu:net1> set activation-mode=prioritized
netcfg:ncp:my_profile:ncu:net1> list
ncu:net1
              type                    link
              class                   phys
              parent                  "my_profile"
              activation-mode         prioritized
              enabled                 true
netcfg:ncp:my_profile:ncu:net1> commit
Committed changes
netcfg:ncp:my_profile:ncu:net1> end
netcfg:ncp:my_profile> exit
```

Listing Reactive Network Profiles

Use the `netcfg` utility to list all the NCPs and locations:

```
# # netcfg list
NCPs:
     Automatic
     DefaultFixed
     my_profile
Locations:
     aces
     Automatic
     NoNet
     DefaultFixed
     office
```

Enabling and Disabling Reactive Network Profiles

Use the `netadm` utility to enable and disable an NCP or a location profile.

- To enable newly created profiles:

```
# netadm enable office
Enabling loc 'office'
# netadm enable my_profile
Enabling ncp 'my_profile
```

- To disable newly created profiles:

```
# netadm disable office
Disabling loc 'office'
# netadm enable -p ncp Automatic
Enabling ncp 'Automatic
```

Displaying Profile States

To list the reactive network profiles and their current states, use the `netadm` utility.

```
# netadm list
TYPE        PROFILE        STATE
ncp         Automatic      disabled
ncp         DefaultFixed   disabled
ncp         my_profile     online
ncu:phys    net0           online
ncu:ip      net0           online
loc         office         online
loc         Automatic      offline
loc         NoNet          offline
loc         DefaultFixed   offline
```

Displaying Profiles and Their Auxiliary States

To list the reactive network profiles and their auxiliary states, use `netadm list -x`.

```
# netadm list -x
TYPE       PROFILE       STATE     AUXILIARY STATE
ncp        Automatic     disabled  disabled by administrator
ncp        DefaultFixed  disabled  disabled by administrator
ncp        my_profile    online    active
ncu:phys   net0          online    interface/link is up
ncu:ip     net0          online    interface/link is up
loc        office        online    active
loc        Automatic     offline   conditions for activation are unmet
Loc        DefaultFixed  offline   conditions for activation are unmet
loc        NoNet         offline   conditions for activation are unmet
```

Creating a Backup of a Profile

To create a backup of a reactive network profile, use `netcfg export -f profile`.

```
# netcfg export -f oracle_ncp_backup ncp my_profile
# ls *backup
oracle_ncp_backup
```

Removing Reactive Network Profiles

To remove a profile, use `netcfg destroy`.

```
# netcfg destroy loc office
# netcfg destroy ncp my_profile
```

Agenda

- Reviewing Networking Fundamentals
- Administering a Datalink Configuration
- Administering a Network Interface
- Administering a Profile-Based Network Configuration
- **Configuring a Virtual Network**
- Verifying Network Operations
- Managing Resources on the Virtual Network

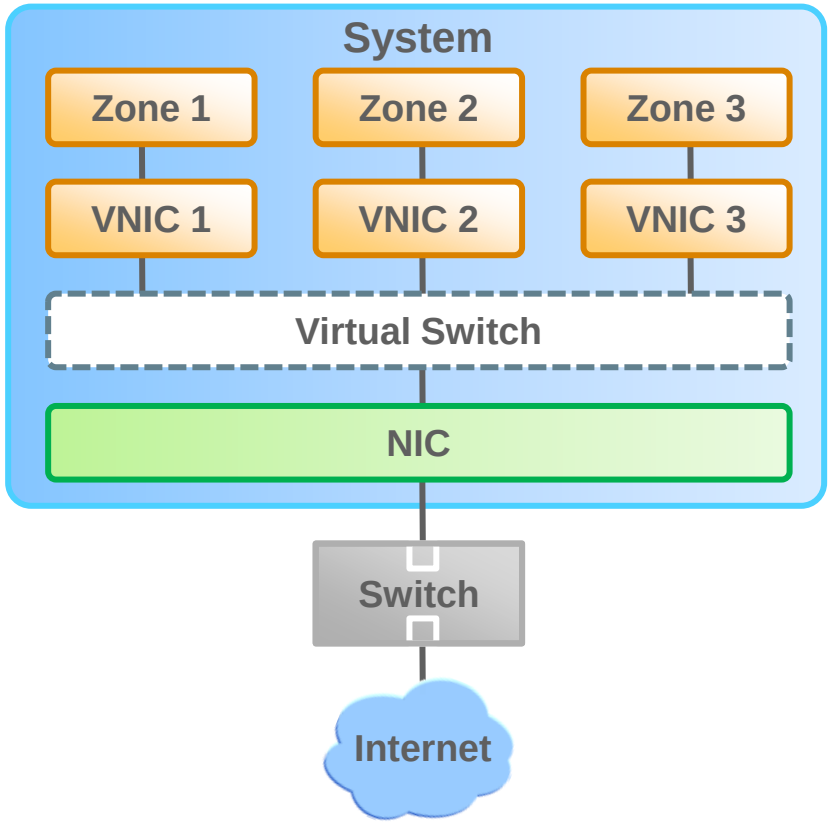
Network Virtualization and Virtual Networks

- Network virtualization:
  - Is the process of combining hardware network resources and software network resources
  - Provides efficient, controlled, and secure sharing of network resources
- Virtual networks:
  - **External networks:** Several local networks that are administered by software as a single entity
  - **Internal networks:** One system that uses virtual machines or zones that are configured over at least one pseudo network interface
    - A special type of internal virtual network is the **private virtual network**, which is a virtual network on a system that cannot be accessed by external networks.

Virtual Network Components

A virtual network has the following components:

- Virtual Network Interface Card (VNIC)
- Virtual switch
- Etherstub
- Zone



Creating a Virtual Network

This section covers the following topics:

- Creating a virtual network switch
- Creating the virtual network interfaces
- Displaying the virtual network configuration

Creating a Virtual Network Switch

To create an etherstub, use `dladm create-etherstub` *`etherstub`*.

```
# dladm create-etherstub stub0
```

To verify the creation of the etherstub, use `dladm show link`.

```
# dladm show-link
LINK        CLASS     MTU    STATE    OVER
net0        phys      1500   up       --
net1        phys      1500   up       --
net2        phys      1500   up       --
net3        phys      1500   up       --
stub0       etherstub 9000   unknown  --
```

Creating the Virtual Network Interfaces

To create a VNIC and attach it to the etherstub, use `dladm create-vnic -l` *`etherstub vnic`*.

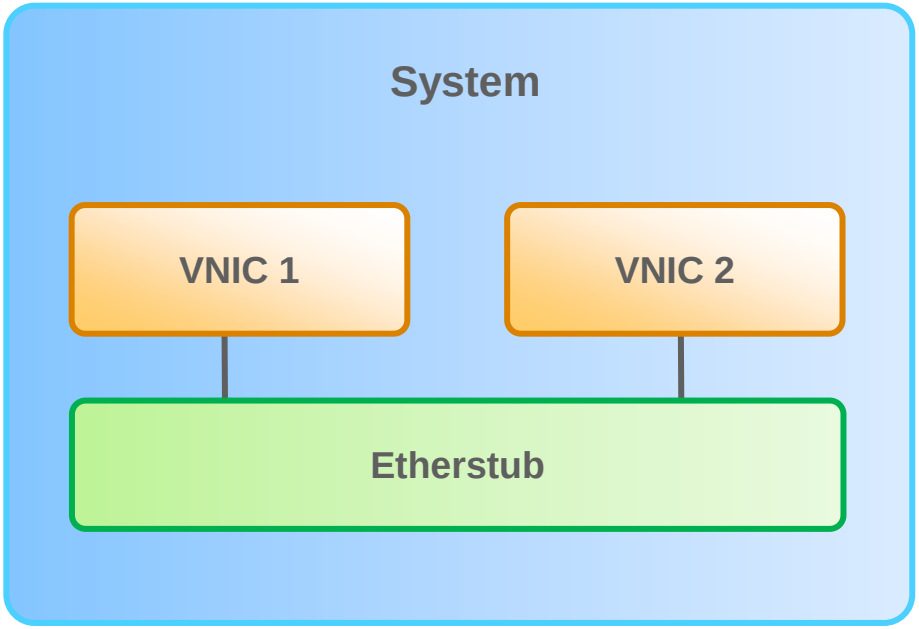
```
# dladm create-vnic -l stub0 vnic0
# dladm create-vnic -l stub0 vnic1
# dladm create-vnic -l stub0 vnic2
```

Displaying the Virtual Network Configuration

To display the virtual network configuration, use `dladm show-vnic`.

```
# dladm show-vnic
LINK        OVER        SPEED  MACADDRESS        MACADDRTYPE   VID
vnic0       stub0       40000  2:8:20:70:d0:f8   random        0
vnic1       stub0       40000  2:8:20:80:65:0    random        0
vnic2       stub0       40000  2:8:20:1f:c5:bd   random        0
```

The Virtual Network Configuration So Far



Quiz

Which utility is used to create virtual switches and VNICs?

a. `lnkadm`

b. `dladm`

c. `vniccfg`

d. `dlcfg`

Quiz

Which utility is used to create virtual switches and VNICs?

a. `lnkadm`

b. `dladm`

c. `vniccfg`

d. `dlcfg`

Quiz

A VNIC is a virtual network device with the same datalink interface as a physical interface.

a. True
b. False

Quiz

A VNIC is a virtual network device with the same datalink interface as a physical interface.

a. True
b. False

Quiz

In which order is a virtual network created?

a. Virtual switch, VNICs, zones
b. Zones, VNICs, virtual switch
c. VNICs, virtual switch, zones

Quiz

In which order is a virtual network created?

a. Virtual switch, VNICs, zones
b. Zones, VNICs, virtual switch
c. VNICs, virtual switch, zones

Quiz

You have created an etherstub called `stub2`. You now want to create `vnic1` and attach it to `stub2`. Which set of commands would you use to do this?

a. `# dladm create-vnic1`
b. `# dladm create-vnic -l vnic1`
c. `# dladm create-vnic -l stub2 vnic0`
d. `# dladm create-vnic -l stub2 vnic1`

Quiz

You have created an etherstub called `stub2`. You now want to create `vnic1` and attach it to `stub2`. Which set of commands would you use to do this?

a. `# dladm create-vnic1`
b. `# dladm create-vnic -l vnic1`
c. `# dladm create-vnic -l stub2 vnic0`
d. `# dladm create-vnic -l stub2 vnic1`

Agenda

- Reviewing Networking Fundamentals
- Administering a Datalink Configuration
- Administering a Network Interface
- Administering a Profile-Based Network Configuration
- Configuring a Virtual Network
- **Verifying Network Operations**
- Managing Resources on the Virtual Network

Verifying Network Operations

- Examining the status of all network interfaces
- Checking network interface traffic status
- Verifying the status of network interfaces
- Checking the routing table
- Viewing user and process information
- Viewing statistics on IP, TCP, and UDP traffic
- Checking network connectivity and response times
- Capturing packets from the network

Examining the Status of All Network Interfaces

To display all the network interfaces, their IP addresses, and status, use `ipadm show-addr`.

```
# ipadm show-addr
ADDROBJ           TYPE     STATE    ADDR
lo0/v4            static   ok       127.0.0.1/8
net0/v4           static   ok       192.168.1.111/24
net1/v4           static   ok       192.168.1.101/24
net2/v4           static   ok       192.168.1.202/24
net3/v4           static   ok       192.168.1.203/24
lo0/v6            static   ok       ::1/128
net0/v6           addrconf ok       fe80::a00:27ff:fe68:6f2d/10
```

Examining the Status of All Network Interfaces

To display network interface configuration information, use `ipadm show-if`.

```
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
net0       ip       ok       yes    --
net1       ip       ok       yes    --
net2       ip       ok       yes    --
net3       ip       ok       yes    --
```

Checking Network Interface Traffic Status

To check network traffic on the network interface, use `netstat -I interface` *`interval count`*.

```
# netstat -I net0 -i 5
     input   net0       output       input   (Total)    output
packets errs  packets errs  colls  packets errs  packets errs  colls
582     0     69      0     0      2732    0     1364    0     0
0       0     0       0     0      0       0     0       0     0
0       0     0       0     0      1       0     2       0     0
1       0     0       0     0      5       0     1       0     0
0       0     0       0     0      0       0     0       0     0
0       0     0       0     0      0       0     0       0     0
^C
```

Verifying the Status of Network Interfaces

To display the status of the network interfaces, use the `netstat -i` command.

```
# netstat -i

Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 loopback      localhost      845037 0     845037 0     0      0
net0  1500 server1       server1        87805  0     126771 0     0      0

...
<output truncated>
```

Checking the Routing Table

To view the known routes, use the `netstat -r` command.

```
# netstat -r
Routing Table:  IPv4
Destination   Gateway     Flags   Ref   Use    Interface
-----------   --------    -----   ---   ----   ---------
localhost     localhost   UH      2     2817   lo0
192.168.1.0   server1     U       4     14293  net0
. . .
<output truncated>
```

Viewing User and Process Information

To list the user, the process ID, and the program that originally created the network endpoint or controls it now, use the `netstat -u` command.

```
# netstat -nauv
UDP: IPv4
   Local Address        Remote Address      User     Pid     State       Command
-------------------- -------------------- -------- ------ ---------- ----------------
      *.*                                  root         79 Unbound    /lib/inet/in.mpathd
      *.*                                  root         79 Unbound    /lib/inet/in.mpathd
      *.*                                  netadm      308 Unbound    /lib/inet/nwamd
      *.*                                  netadm      308 Unbound    /lib/inet/nwamd
      *.631                                root        430 Idle       /usr/sbin/cupsd -C /etc/cups/cupsd.conf
127.0.0.1.53                               root        443 Idle       /usr/sbin/named
192.168.1.200.53                           root        443 Idle       /usr/sbin/named
      *.111                                daemon      539 Idle       /usr/sbin/rpcbind
      *.*                                  daemon      539 Unbound    /usr/sbin/rpcbind
      *.52951                              daemon      539 Idle       /usr/sbin/rpcbind
      *.111                                daemon      539 Idle       /usr/sbin/rpcbind
      *.*                                  daemon      539 Unbound    /usr/sbin/rpcbind
      *.36871                              daemon      539 Idle       /usr/sbin/rpcbind
      *.*                                  root        585 Unbound    /usr/lib/inet/in.ndpd
      *.520                                root        782 Idle       /usr/sbin/in.routed
      *.68                                 root        787 Idle       /sbin/dhcpagent
      *.546                                root        787 Idle       /sbin/dhcpagent
. . .
<output truncated>
```

Viewing Statistics on IP Traffic

To gather and report statistics on IP traffic based on a selected output mode and sort order, use the `ipstat` command.

```
# ipstat -l 5
SOURCE                          DEST                            PROTO    INT         BYTES
s11-server1.example.com         s11-desktop.example.com         UDP      net0        39.0
s11-desktop.example.com         s11-server1.example.com         UDP      net0        28.0
Total: bytes in: 39.0  bytes out: 28.0
```

Viewing Statistics on TCP and UDP Traffic

To gather and report statistics on TCP and UDP traffic based on a selected output mode and sort order, use the `tcpstat` command.

```
# tcpstat -l 5
ZONE          PID PROTO   SADDR               SPORT DADDR               DPORT   BYTES
global        795 UDP     s11-server1.exam       53 s11-desktop.exam   42857    20.0
global        795 UDP     s11-desktop.exam    42857 s11-server1.exam       53     9.0
global        795 UDP     s11-desktop.exam    59127 s11-server1.exam       53     7.0
global        795 UDP     s11-desktop.exam    38509 s11-server1.exam       53     7.0
global        795 UDP     s11-server1.exam       53 s11-desktop.exam   59127     7.0
Total: bytes in: 34.0  bytes out: 23.0
```

Checking Network Connectivity and Response Times

To check connectivity between one host and another, use the `ping` command.

```
# ping -s 192.168.1.112
PING 192.168.1.112: 56 data bytes
64 bytes from s11-serv1.example.com (192.168.1.112): icmp_seq=0. time=1.143 ms
64 bytes from s11-serv1.example.com (192.168.1.112): icmp_seq=1. time=0.724 ms
64 bytes from s11-serv1.example.com (192.168.1.112): icmp_seq=2. time=1.639 ms
^C
----192.168.1.112 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms) min/avg/max/stdev = 1.639/0.724/1.143/0.649
```

Capturing Packets from the Network

To capture packets, use `snoop`.

```
# snoop -v
Using device net0 (promiscuous mode)
ETHER:  ----- Ether Header -----
ETHER:  Packet 1 arrived at 13:52:2.50694
ETHER:  Packet size = 106 bytes
ETHER:  Destination = 0:7:e9:24:45:93, PCS Computer Systems GmbH
ETHER:  Source      = 0:3:ba:45:a6:d4,
ETHER:  Ethertype = 0800 (IP)
. . . . .
IP:   ----- IP Header -----
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
. . . . .
^C
```

Quiz

Which command can you use to display your system's current network interface configuration?

a. `ipadm`

b. `ping`

c. `netstat -I`

Quiz

Which command can you use to display your system's current network interface configuration?

a. `ipadm`

b. `ping`

c. `netstat -I`

Agenda

- Reviewing Networking Fundamentals
- Administering a Datalink Configuration
- Administering a Network Interface
- Administering a Profile-Based Network Configuration
- Configuring a Virtual Network
- Verifying Network Operations
- **Managing Resources on the Virtual Network**

Network Resource Management: Overview

- Network resource management is the process of managing and allocating resources for networking processes.
- It is comparable to creating dedicated lanes for traffic.
- You can assign resources differently depending on the amount of network traffic that is being processed.
- It helps in increasing a system's efficiency when processing packets.

Methods of Managing Network Resources

The network resources on a system can be managed in one of the following ways:

- **Datalink properties:** Improves the system's efficiency in processing packets
- **Flows:** Controls how resources are used to process network packets

Managing Virtual Network Resources by Using Flows

Flows:

- Are created on a per-NIC or per-VNIC basis
- Are used to categorize network packets
- Define and isolate packets with similar characteristics
- Can be assigned specific resources

Bandwidth is assigned based on the usage policy for the system.

Managing Resources on the Virtual Network

This section covers the following topics:

- Determining the configured VNIC states
- Creating and adding a flow to a VNIC
- Displaying flow controls
- Creating flows and selecting flow properties
- Setting flow properties
- Displaying flow control properties
- Setting a priority property

Determining the Configured VNIC States

To determine the current state of the VNICs on the system, use `dladm show-link`.

```
# dladm show-link
LINK              CLASS         MTU      STATE    OVER
net0              phys          1500     up       --
stub0             etherstub     9000     unknown  --
net3              phys          1500     up       --
net1              phys          1500     up       --
net2              phys          1500     up       --
vnic0             vnic          9000     up       stub0
vnic1             vnic          9000     up       stub0
vnic2             vnic          9000     up       stub0
```

Creating and Adding a Flow

1. Create a new VNIC by using `dladm create-vnic -l` *`etherstub vnic`*.
2. Select the attribute on which you want to base the flow.
3. Determine how you want to customize the flow's use of the network resources.
4. Add the VNIC as a flow by using `flowadm add-flow -l link -a attribute=value` *`flow`*.

```
# dladm create-vnic -l stub0 vnic3
# flowadm add-flow -l vnic3 -a transport=tcp,local_port=80 http1
```

Displaying Flow Controls

To display the flow controls that are currently configured in the system, use `flowadm show-flow`.

```
# flowadm show-flow
FLOW     LINK      PROTO LADDR          LPORT RADDR              RPORT DIR
http1    vnic3     tcp   --             80    --                 --    bi
```

Creating Flows and Selecting Flow Properties

- Flows are created according to attributes.
- Attributes are classifications that are used to organize network packets into a flow.
- Flows use properties to control resources:
  - **`maxbw`:** Maximum amount of a link's bandwidth that packets identified with this flow can use
  - **`priority`:** Priority given to the packets in a flow:
    - Options: `high`, `medium`, or `low`
    - Default: `medium`

Setting Flow Properties

To set a flow property, use `flowadm set-flowprop -p property=value` *flow*.

```
# flowadm set-flowprop -p maxbw=100M http1
```

Displaying Flow Control Properties

To display a flow's control properties, use `flowadm show-flowprop flow`.

```
# flowadm show-flowprop http1
FLOW      PROPERTY      PERM VALUE     DEFAULT    POSSIBLE
http1     maxbw         rw   100       --         --
http1     priority      rw   medium    medium     low,medium,high
http1     hwflow        r-   off       --         on,off
```

Setting a Priority Property

To set a link property, use `dladm set-linkprop -p property=high` *`vnic`*.

```
# dladm set-linkprop -p priority=high vnic1
```

To view the priority property for a link, use `dladm show-linkproperty -p priority` *`vnic`*.

```
# dladm show-linkprop -p priority vnic1
LINK      PROPERTY      PERM VALUE    EFFECTIVE    DEFAULT    POSSIBLE
vnic1     priority      rw   high     high         medium     low,medium,high
```

Quiz

Which two properties do flows use to control resources?

1. `speed` and `mtu`
2. `maxbw` and `priority`
3. `flowctrl` and `threshold`

Quiz

Which two properties do flows use to control resources?

1. `speed` and `mtu`
2. `maxbw` and `priority`
3. `flowctrl` and `threshold`

Summary

In this lesson, you should have learned how to:

- Describe some of the basic networking concepts
- Administer a datalink configuration
- Administer a network interface
- Administer a profile-based network configuration
- Configure a virtual network
- Verify network operations
- Manage resources on the network

Practices 7: Overview

- 7-1: Manually Configuring the Network Interface
- 7-2: Administering a Profile-Based Network Configuration
- 7-3: Creating a Virtual Network
- 7-4: Verifying Network Operations
- 7-5: Managing a Virtual Network Data Flow