



Information Sciences Institute

NetFS – Network Configuration through the File System

Joshua Train, Joseph D. Touch

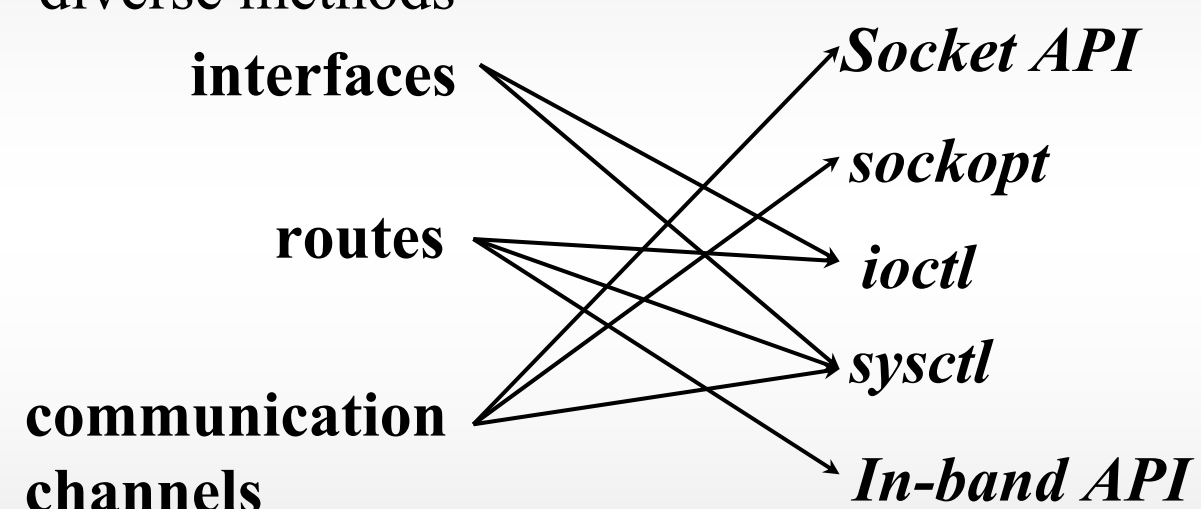
Limitations of Current Configuration Methods

All or None Access

- Network configuration requires root access
 - Multiple VPN's require Individual Application/ User Network Configurations
 - X-Bone – for overlay applications
 - PlanetLab – for VPN configuration
- Root is too “strong” to be given out to multiple users.

Portability

- Network Configuration is performed by various diverse methods



- Syntax varies between flavors of UNIX
- Different operating systems have different methods

NetFS Provides

Fine-Grained Access Control

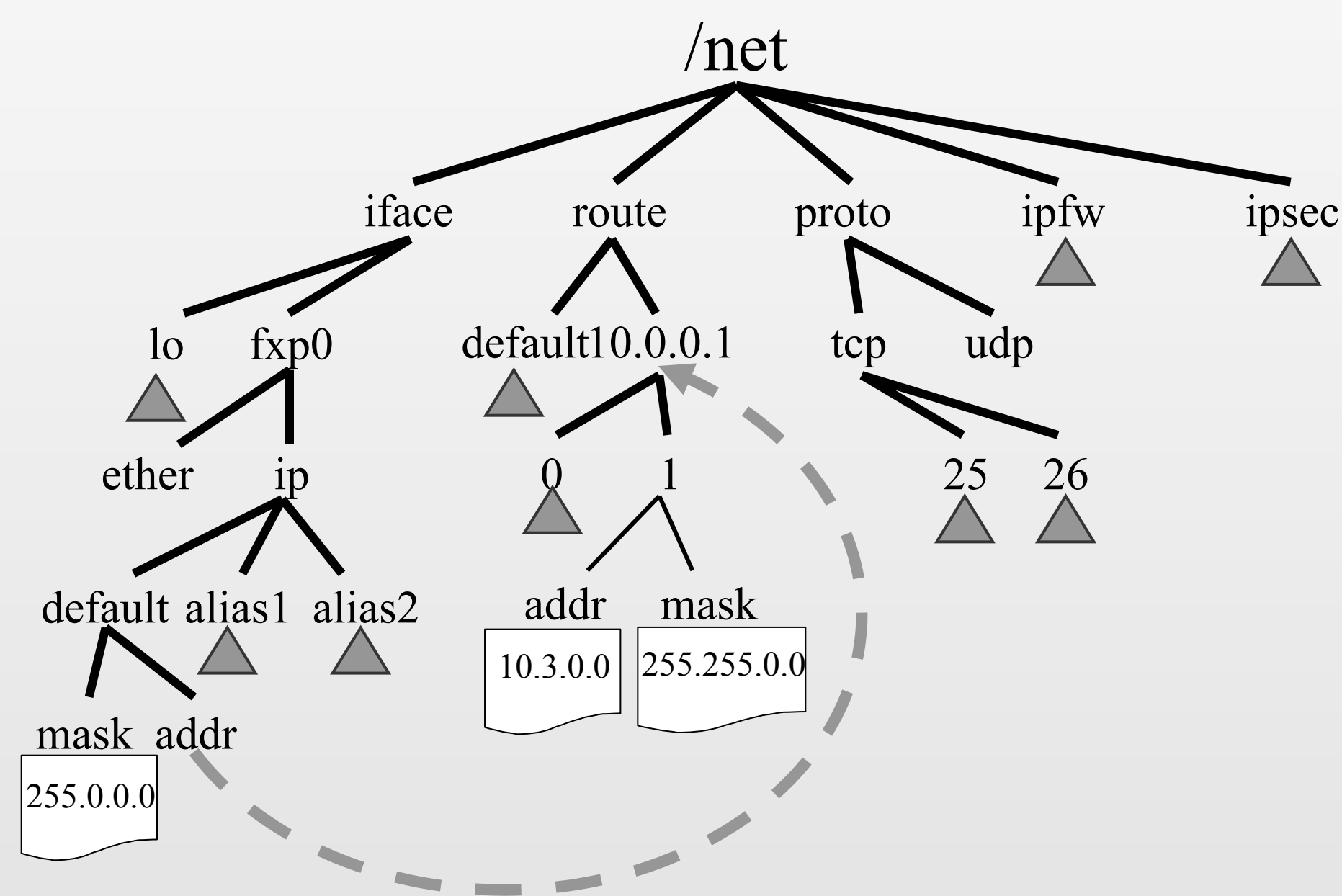
- Access control via file permissions
- Supports user/process level granularity
- Hierarchical permissions structure

Familiar, Unified Interface

- File system operations common to users
- File system operations are standard between operating systems.
 - VNODE standard
 - NFS
- Same operations for different components
- Similar to ProcFS

Description

- Instances of networking components are described by directories
 - Interfaces contained in /net/ifaces directory
 - Routes contained in /net/routes directory
- Attributes are stored in files
- Configuration is performed by file system operations
 - Listing contents of directories
 - Creating/removing directories
 - Writing/reading files



Implementation

- Implemented in Free BSD 5.2
 - Kernel Patches
- Currently Implemented:
 - Routes
 - Interfaces
 - Bind limited based on IP address permissions
- Planned configuration support for:
 - Sockets, IPsec, firewalls

Examples

Viewing Information

Viewing Interfaces:
`ls /net/ifaces`
 Output:
 10 xl0 em0

Viewing Routes:
`ls /net/routes`
 Output:
 Default 192.168.0.0 10.0.0.0

Reading Gateway Information:
`cat /net/routes/default/gateway`
 Output:
 192.168.1.1

Creating or Removing Components

Creating Interfaces:
`mkdir /net/ifaces/gif0`

Removing Interfaces:
`rmdir /net/ifaces/gif0`

Creating a route:
`mkdir /net/routes/192.168.0.0`

Removing a route:
`rmdir /net/routes/192.168.0.0`

Setting Permissions on Components

Changing ownership of an interface:
`chown newuser /net/ifaces/gif0`

Changing permission on a route:
`chmod 744 /net/routes/default`

Future Opportunities

- Remote network configuration
 - NFS mounts
- Aggregate configuration
- Aggregate consistency control
- Network virtualization
- Cross operating system compatibility