nf.io: A File System Abstraction for NFV Orchestration

Md. Faizul Bari, Shihabur R. Chowdhury, Reaz Ahmed, and Raouf Boutaba
David R. Cheriton School of Computer Science
University of Waterloo
Outline

- What is nf.io?
- Related works
- Why File System abstraction?
- The nf.io File System Abstraction
- Sample Use cases
- System Architecture
- Prototype Implementation
- Work in Progress
- Conclusion
What is nf.io?

NFV Users / Operators

Northbound API for Deploying, Configuring, and Monitoring Virtual Network Functions (VNFs)
RESTful? Policy Language? other?

NFV Management & Orchestration

Southbound API:
OpenStack nova/neutron, libvirt, OpenFlow etc.

Compute
Network
VNFs

NFV Infrastructure

* Simplified view of ETSI Reference NFV Architecture
What is nf.io?

NFV Users / Operators

Northbound API for Deploying, Configuring, and Monitoring Virtual Network Functions (VNFs)
Our Proposal: nf.io

NFV Management & Orchestration

Southbound API:
OpenStack nova/neutron, libvirt, OpenFlow etc.

NFV Infrastructure

* Simplified view of ETSI Reference NFV Architecture
What is nf.io?

nf.io: A Northbound Interface for NFV Management & Orchestration

NFV Management & Orchestration
What is nf.io?

nfio: A Northbound Interface for NFV Management & Orchestration

NFV Management & Orchestration

“Everything” (VNF/configuration/state) is represented in a file system hierarchy

nfio

user-a

nf-types

firewall

fw-instance

ids

ids-instance

nf.io

Compute

Network

VNFs

NFV Infrastructure
What is nf.io?

nf.io: A Northbound Interface for NFV Management & Orchestration

NFV Management & Orchestration

“Everything” (VNF/configuration/state) is represented in a file system hierarchy

nf.io

Compute
Network
VNFs

NFV Infrastructure

Deploy NF

nf.io

... 

Compute Driver

- Infrastructure agnostic
  high level operations
- Resource specific
drivers

docker
libvirt
What is nf.io?

 nfio: A Northbound Interface NFV Management & Orchestration

“Everything” (VNF/configuration/state) is represented in a file system hierarchy

NFV Infrastructure

Deploy NF

nf.io

Compute Driver
docker libvirt

- Infrastructure agnostic high level operations
- Resource specific drivers
Related Work

● NFV Platforms
  ○ Stratos, OpenNF [SIGCOMM ‘14]
  ○ E2 [SOSP ‘15]
  ○ OPNFV

● File System Abstraction
  ○ Linux sysfs, procfs, cgroup
  ○ yanc [HotNets ‘13]

● Cloud Orchestrator
  ○ OpenStack, CloudStack, SaltStack
Why File System Abstraction?

- Familiar tools to manage file systems
  - `mkdir`, `cp`, `move`, `rm`, `rsync`, etc.
  - `grep`, `sed`, `awk`, `tail`, etc.
    - e.g., instantiate a new VNF
    - `mkdir -p /vnfs/user-a/chain-b/ids`

- Rich set of file system management operations offered by configuration management tools such as Chef, Puppet, Salt etc.
Why File System Abstraction?

- Familiar tools to manage file systems
  - `mkdir, cp, move, rm, rsync, etc.`
  - `grep, sed, awk, tail, etc.`
    - e.g., instantiate a new VNF
      - `mkdir -p /vnfs/user-a/chain-b/ids`
- Rich set of file system management operations offered by configuration management tools such as Chef, Puppet, Salt etc.

Rich toolset around File Systems can be leveraged if we can translate NFV operations to File System Operations
nf.io: File System Abstraction: High level

```
nfio
  └── user-a
      ├── chns
      │    └── nf-types
      │         ├── fw
      │         │    └── lb
      │         └── wp
      └── user-b
```

nf.io root
nf.io: File System Abstraction: High level

- `nfio`
- `user-a`
  - `chns`
  - `nf-types`
    - `fw`
    - `lb`
    - `wp`
- `user-b`

Per user directory for VNFs and chains
nf.io: File System Abstraction: High level

Contains VNF chains
nf.io: File System Abstraction: High level

Contains VFS, categorized into types

Contains VNF instances and Common Configurations
nf.io File System Abstraction: A Sample VNF

A Specific Firewall Instance
nf.io File System Abstraction: A Sample VNF

Writing “start” / “stop” here will “start” or “stop” the Firewall
nf.io File System Abstraction: A Sample VNF

Configuration of the Firewall’s container/VM
nf.io File System Abstraction: A Sample VNF

Mount point for the remote file system of the container/VM
nf.io File System Abstraction: A Sample VNF

- action
- config
  - boot.conf
- machine
  - ip
  - vm.memory
  - vm.vcpu
- rfs
  - rules
    - 1
      - action
        - match
          - matched_bytes
          - matched_pkts
  - stats
    - pkt_drops
    - rx_bytes
    - tx_bytes
    - status

Firewall specific: Represents firewall rules
Action to take, e.g., allow/drop
Headers to match
nf.io File System Abstraction: A Sample VNF

- fw
  - fw1
    - action
    - config
      - boot.conf
    - machine
      - ip
      - vm.memory
      - vm.vcpu
    - rfs
    - rules
      - 1
        - action
        - match
          - matched_bytes
          - matched_pkts
    - stats
      - pkt_drops
      - rx_bytes
      - tx_bytes

Placeholder virtual files for reading counters
nf.io File System Abstraction: A VNF Chain

Example:

```
+----------------+            +----------------+
| Firewall       | ➤                | Proxy           |
```

```
chns
   | chain-alpha
   |   | action
   |   | status
   |   | strand-1 ➤ Represents branches of a chain
   |   |   | fw-alpha ➤ /vnfs/user-a/nf-types/fw/fw-alpha
   |   |   |   | next ➤ /vnfs/user-a/nf-types/proxy/proxy-alpha
   |   | proxy-alpha ➤ /vnfs/user-a/nf-types/proxy/proxy-alpha
   |   |   | start ➤ /vnfs/user-a/nf-types/fw/fw-alpha
   | chain-beta
```
nf.io File System Abstraction: A VNF Chain

Example:

```
+-------------------+        +-------------------+
| Firewall          |        | Proxy             |
+-------------------+        +-------------------+
```

```
+-------------------+        +-------------------+
| chns              |        | One directory for each chain component |
+-------------------+        +-------------------+
| chain-alpha       |        |                   |
|    +-------------+        +------------------+
|    | action       |        |                   |
|    +-------------+        +------------------+
|    | status       |        |                   |
|    +-------------+        +------------------+
| strand-1        |        |                   |
|       +-----------+        +------------------+
|       | fw-alpha    |        |                   |
|       +-----------+        +------------------+
|   | fw-alpha -> /vnfs/user-a/nf-types/fw/fw-alpha |
|   +-----------+        +------------------+
|   | next -> /vnfs/user-a/nf-types/proxy/proxy-alpha |
|   +-----------+        +------------------+
| proxy-alpha   |        |                   |
|       +-----------+        +------------------+
|       | proxy-alpha -> /vnfs/user-a/nf-types/proxy/proxy-alpha |
|       +-----------+        +------------------+
|       | start -> /vnfs/user-a/nf-types/fw/fw-alpha |
|       +-----------+        +------------------+
| chain-beta     |        |                   |
```
nf.io File System Abstraction: A VNF Chain

Example: Firewall -> Proxy

```
chns
  chain-alpha
    action
    status
    strand-1
      fw-alpha
        fw-alpha -> /vnfs/user-a/nf-types/fw/fw-alpha
        next   -> /vnfs/user-a/nf-types/proxy/proxy-alpha
      proxy-alpha
        proxy-alpha -> /vnfs/user-a/nf-types/proxy/proxy-alpha
      start     -> /vnfs/user-a/nf-types/fw/fw-alpha
```
nf.io File System Abstraction: A VNF Chain

Example:

```
Example: Firewall -> Proxy
```

```
<table>
<thead>
<tr>
<th>chns</th>
<th>chain-alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>action</td>
</tr>
<tr>
<td></td>
<td>status</td>
</tr>
<tr>
<td>strand-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fw-alpha</td>
</tr>
<tr>
<td></td>
<td>proxy-alpha</td>
</tr>
</tbody>
</table>

Symbolic link to the first VNF instance in the chain
Sample Use Cases

- **Deploy a VNF instance**
  - `mkdir nf-types/bro/ids-a`

- **Configure a VNF instance**
  - Migrate a VNF instance to a different machine
    `echo '10.0.0.15' > chns/chain-alpha/fw-alpha/fw-a/machine/ip`

- **Monitor**
  - Total number of packet drops along a chain:
    `find -L chns/chain-a pkt_drops | xargs cat | awk '{total += $1} END {print total}’`
System Architecture
Prototype Implementation

nf.io File System

- Command Line Utils.
- Custom Scripts
- Automation Tools

- Hypervisor Driver
  - Compute Resources
- Network Driver
  - Network Resources
- Chain Driver
  - VNF Chaining

User space filesystem using **FUSE** (File System in User Space) python library
Prototype Implementation

nf.io File System

- **Command Line Utils.**
- **Custom Scripts**
- **Automation Tools**

**Hypervisor Driver**
- Compute Resources

**Network Driver**
- Network Resources

**Chain Driver**
- VNF Chaining

Docker python API

Docker containers on physical servers

User space filesystem using **FUSE (File System in User Space)** python library
Prototype Implementation

nf.io File System

- Command Line Utils
- Custom Scripts
- Automation Tools

- Hypervisor Driver
- Network Driver
- Chain Driver

- Docker python API
- Docker containers on physical servers

User space filesystem using FUSE (File System in User Space) python library

- Compute Resources
- Network Resources
- VNF Chaining
- Linux bridge, route, iptables

End-host networking only
Prototype Implementation

nf.io File System

Command Line Utils.      Custom Scripts      Automation Tools

Hypervisor Driver

Network Driver

Chain Driver

Docker python API

Docker containers on physical servers

Compute Resources

Network Resources

End-host networking only

Linux bridge, route, iptables

User space filesystem using FUSE (File System in User Space) python library

- nginx as proxy
- bro IDS
- iptables firewall
Work In Progress

- Add support for VNF chains with branches
- Simplify VNF chain representation
- Open vSwitch for end-host data-path
  - Better Programmability
- Add more hypervisor support
  - Xen, KVM
- Integrate with OpenStack
Conclusion

- **nf.io**: A northbound interface for interacting with VNF Management and Orchestration Systems
- **nf.io** adapts the well known Linux File System abstraction
- **nf.io** allows high level infrastructure agnostic VNF operations
- Live demo of **nf.io**:
  - [http://faizulbari.github.io/nf.io/](http://faizulbari.github.io/nf.io/) (Link is also in the paper)
  - Try it out!
Questions?