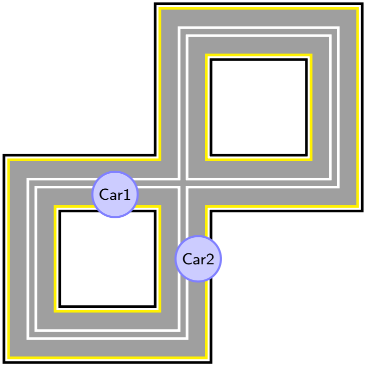
Conflict Resolution in SDN: Over-viewing Approaches and ONOS Intent Framework Usage on a Study Case

Manassés Ferreira

Intelligent Networks, PPGCC, ICEx, UFMG

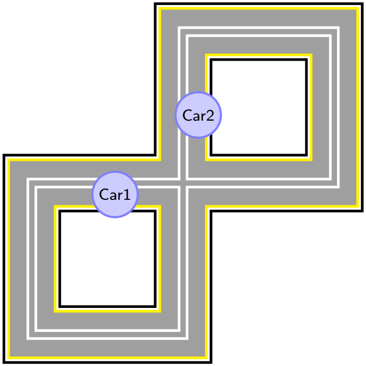
December 21, 2015

What is Conflict? How to Model? How to Resolve?



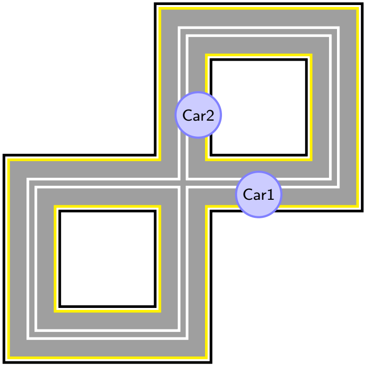
Two Cars need to use the same crossroad.

What is Conflict? How to Model? How to Resolve?



Who is on the right should go first.

What is Conflict? How to Model? How to Resolve?



A simple rule ends the conflict.

Conflict Resolution in SDN

RESERVE *n*

App1 wants RESERVE 10 and App2 asks for RESERVE 30.

Conflict Resolution in SDN

RESERVE *n*

App1 wants RESERVE 10 and App2 asks for RESERVE 30.

WAYPOINT / AVOID

App1 intends to WAYPOINT *IP* and App2 desires AVOID the same *IP*.

Conflict Resolution in SDN

RESERVE *n*

App1 wants RESERVE 10 and App2 asks for RESERVE 30.

WAYPOINT / AVOID

App1 intends to WAYPOINT *IP* and App2 desires AVOID the same *IP*.

DENY/ALLOW

App1 may wish to DENY traffic to TCP port 80, while another App2 may require such traffic be ALLOWed.

Conflict Resolution in SDN

Reserve *n*

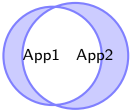
App1 wants Reserve 10 and App2 asks for Reserve 30.

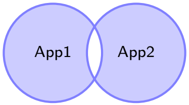
Waypoint / Avoid

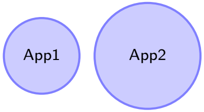
App1 intends to Waypoint *IP* and App2 desires Avoid the same *IP*.

Deny/Allow

App1 may wish to Deny traffic to TCP port 80, while another App2 may require such traffic be Allowed.







Two Apps overlap when the intersection of their respective demands is not empty, i.e., there are some flows that match both.

Over-viewing Approaches

PANE [FGL+13]

Provides a way for the network to solicit and react to such conflict needs automatically, dynamically, and at a finer timescale than with human input.
Participatory Networking, **Policy Atoms**, **Policy Tree**

Over-viewing Approaches

PANE [FGL+13]

Provides a way for the network to solicit and react to such conflict needs automatically, dynamically, and at a finer timescale than with human input.
Participatory Networking, **Policy Atoms**, **Policy Tree**

ONOS - Intent Framework [OnL14]

is a subsystem that allows applications to specify their network control desires in form of policy rather than mechanism.
Open Network Operating System, **Intent Compilation**, **Intent States**

Over-viewing Approaches

PANE [FGL+13]

Provides a way for the network to solicit and react to such conflict needs automatically, dynamically, and at a finer timescale than with human input.
Participatory Networking, **Policy Atoms**, **Policy Tree**

ONOS - Intent Framework [OnL14]

is a subsystem that allows applications to specify their network control desires in form of policy rather than mechanism.
Open Network Operating System, **Intent Compilation**, **Intent States**

PGA [PLT+15]

Network policies to be expressed simply and independently, and leverage the graph structure to detect and resolve policy conflicts efficiently.
Policy Graph Abstraction, **Conflict-free Composed Chains**

Intent Framework Based on [OnL14]

- Policy rather than Mechanism

[1] actionable operations on the network environment

[2] such as tunnel links being provisioned, flow rules being installed on a switch, or optical lambdas (wavelengths) being reserved

Intent Framework Based on [OnL14]

- Policy rather than Mechanism
- Intents are compiled into **installable intents** [1]

[1] actionable operations on the network environment

[2] such as tunnel links being provisioned, flow rules being installed on a switch, or optical lambdas (wavelengths) being reserved

Intent Framework Based on [OnL14]

- Policy rather than Mechanism
- Intents are compiled into **installable intents** [1]
- Installation process results in some **changes** [2] to the environment

[1]actionable operations on the network environment

[2]such as tunnel links being provisioned, flow rules being installed on a switch, or optical lambdas (wavelengths) being reserved

Intent Framework Based on [OnL14]

- Policy rather than Mechanism
- Intents are compiled into **installable intents** [1]
- Installation process results in some **changes** [2] to the environment
- Suite of available built-in intents (their compilers and installers)

[1] actionable operations on the network environment

[2] such as tunnel links being provisioned, flow rules being installed on a switch, or optical lambdas (wavelengths) being reserved

Intent Framework Based on [OnL14]

- Policy rather than Mechanism
- Intents are compiled into **installable intents** [1]
- Installation process results in some **changes** [2] to the environment
- Suite of available built-in intents (their compilers and installers)
- Designed to be extensible

[1]actionable operations on the network environment

[2]such as tunnel links being provisioned, flow rules being installed on a switch, or optical lambdas (wavelengths) being reserved

Intent Framework Based on [OnL14]

- Policy rather than Mechanism
- Intents are compiled into **installable intents** [1]
- Installation process results in some **changes** [2] to the environment
- Suite of available built-in intents (their compilers and installers)
- Designed to be extensible
- Intents can be added dynamically at run-time

[1] actionable operations on the network environment

[2] such as tunnel links being provisioned, flow rules being installed on a switch, or optical lambdas (wavelengths) being reserved

Intent Based on [OnL14]

Definition

Immutable model object that describes an application's request to the ONOS core to alter the network's behavior.

Intent Based on [OnL14]

Definition

Immutable model object that describes an application's request to the ONOS core to alter the network's behavior.

Composition

Network Resource	set of object models affected by an intent

Intent Based on [OnL14]

Definition

Immutable model object that describes an application's request to the ONOS core to alter the network's behavior.

Composition

Network Resource	set of object models affected by an intent
Constraints	weights applied to a set of network resources

Intent Based on [OnL14]

Definition

Immutable model object that describes an application's request to the ONOS core to alter the network's behavior.

Composition

Network Resource	set of object models affected by an intent
Constraints	weights applied to a set of network resources
Criteria	patterns that describe a slice of traffic

Intent Based on [OnL14]

Definition

Immutable model object that describes an application's request to the ONOS core to alter the network's behavior.

Composition

Network Resource	set of object models affected by an intent
Constraints	weights applied to a set of network resources
Criteria	patterns that describe a slice of traffic
Instructions	actions to apply to a slice of traffic

Intent Based on [OnL14]

Definition

Immutable model object that describes an application's request to the ONOS core to alter the network's behavior.

Composition

Network Resource	set of object models affected by an intent
Constraints	weights applied to a set of network resources
Criteria	patterns that describe a slice of traffic
Instructions	actions to apply to a slice of traffic

Intent Based on [OnL14]

Definition

Immutable model object that describes an application's request to the ONOS core to alter the network's behavior.

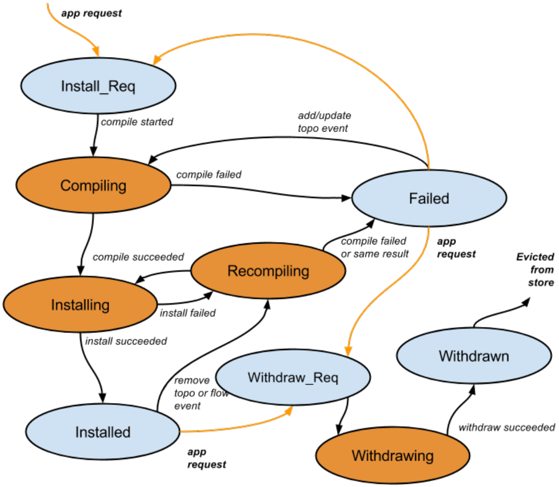
Composition

Network Resource	set of object models affected by an intent
Constraints	weights applied to a set of network resources
Criteria	patterns that describe a slice of traffic
Instructions	actions to apply to a slice of traffic

Identification

Identified by both the **ApplicationId** of the application that submitted it, and a unique **IntentId**, generated at creation.

Intent States Based on [OnL14]



Intent General Operation Based on [OnL14]

- Intent States: Submission [3] $\rightarrow$ Compiling $\rightarrow$ Installing $\rightarrow$ Installed

[3] Sent immediately but asynchronously. The Intent framework relies on the Java Future for handling the asynchronous intent compilation process

Intent General Operation Based on [OnL14]

- Intent States: Submission [3] $\rightarrow$ Compiling $\rightarrow$ Installing $\rightarrow$ Installed
- Withdrawn if no longer wishes for the intent to hold

[3]Sent immediately but asynchronously. The Intent framework relies on the Java Future for handling the asynchronous intent compilation process

Intent General Operation Based on [OnL14]

- Intent States: Submission [3] $\rightarrow$ Compiling $\rightarrow$ Installing $\rightarrow$ Installed
- Withdrawn if no longer wishes for the intent to hold
- Failures by issues that may arise:

[3]Sent immediately but asynchronously. The Intent framework relies on the Java Future for handling the asynchronous intent compilation process

Intent General Operation Based on [OnL14]

- Intent States: Submission [3] $\rightarrow$ Compiling $\rightarrow$ Installing $\rightarrow$ Installed
- Withdrawn if no longer wishes for the intent to hold
- Failures by issues that may arise:
  - Compiling phase may fail
    Connectivity across to unconnected network segments

[3]Sent immediately but asynchronously. The Intent framework relies on the Java Future for handling the asynchronous intent compilation process

Intent General Operation Based on [OnL14]

- Intent States: Submission [3] $\rightarrow$ Compiling $\rightarrow$ Installing $\rightarrow$ Installed
- Withdrawn if no longer wishes for the intent to hold
- Failures by issues that may arise:
  - Compiling phase may fail
    Connectivity across to unconnected network segments
  - Transition back to the compiling state
    Change in the environment

[3] Sent immediately but asynchronously. The Intent framework relies on the Java Future for handling the asynchronous intent compilation process

Intent General Operation Based on [OnL14]

- Intent States: Submission [3] $\rightarrow$ Compiling $\rightarrow$ Installing $\rightarrow$ Installed
- Withdrawn if no longer wishes for the intent to hold
- Failures by issues that may arise:
  - Compiling phase may fail
    Connectivity across to unconnected network segments
  - Transition back to the compiling state
    Change in the environment
  - Recompile the intent to see if an alternate approach is available
    Installing phase may be disrupted

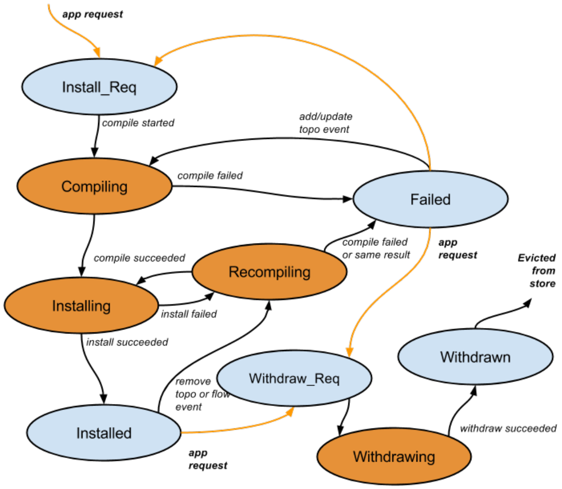
[3]Sent immediately but asynchronously. The Intent framework relies on the Java Future for handling the asynchronous intent compilation process

Intent General Operation Based on [OnL14]

- Intent States: Submission [3] $\rightarrow$ Compiling $\rightarrow$ Installing $\rightarrow$ Installed
- Withdrawn if no longer wishes for the intent to hold
- Failures by issues that may arise:
  - Compiling phase may fail
    Connectivity across to unconnected network segments
  - Transition back to the compiling state
    Change in the environment
  - Recompile the intent to see if an alternate approach is available
    Installing phase may be disrupted
  - Impact the viability of a successfully compiled and installed intent
    Loss of throughput or connectivity

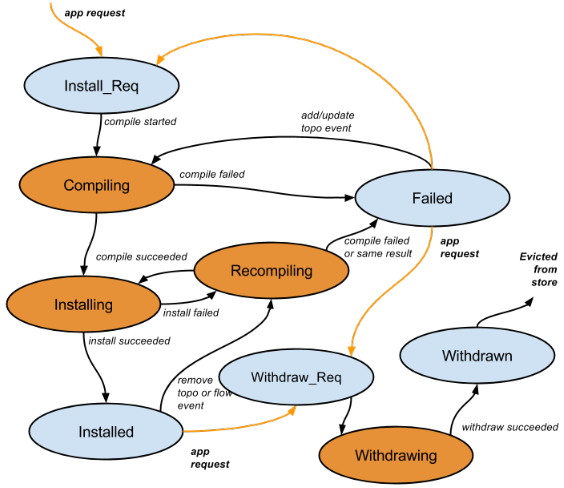
[3] Sent immediately but asynchronously. The Intent framework relies on the Java Future for handling the asynchronous intent compilation process

Intent Compilation Based on [OnL14]



- States depicted in orange are transitional (brief amount of time)

Intent Compilation Based on [OnL14]



- States depicted in orange are transitional (brief amount of time)
- The rest of the states are parking states (more time is spent there)

Intent Compilers and Installers Based on [OnL14]

Intents are ultimately compiled down into a set of **FlowRule** model objects.

Intent Compilers and Installers Based on [OnL14]

Intents are ultimately compiled down into a set of **FlowRule** model objects. The process may include:

1. The compilation of an Intent down into installable intent(s), by an **IntentCompiler**
2. The conversion of installable Intents into **FlowRuleBatchOperations** containing **FlowRules**, by an **IntentInstaller**

Intent Compilers and Installers Based on [OnL14]

Intents are ultimately compiled down into a set of **FlowRule** model objects. The process may include:

1. The compilation of an Intent down into installable intent(s), by an **IntentCompiler**
2. The conversion of installable Intents into **FlowRuleBatchOperations** containing **FlowRules**, by an **IntentInstaller**

The **IntentManager** coordinates the compilation and installation of **FlowRules** by managing the invocation of available **IntentCompilers** and **IntentInstallers**

Intent FlowRule Installation Based on [OnL14]

FlowRuleBatchOperations are handed off to the **FlowRuleManager** to be written down as protocol-specific messages.

Intent FlowRule Installation Based on [OnL14]

FlowRuleBatchOperations are handed off to the **FlowRuleManager** to be written down as protocol-specific messages.

In the case of *OpenFlow*, the *OpenFlowRuleProvider* is responsible for generating *OFFlowMod* messages from **FlowRules**, and sending it to the appropriate network switch.

Intent FlowRule Installation Based on [OnL14]

FlowRuleBatchOperations are handed off to the **FlowRuleManager** to be written down as protocol-specific messages.

In the case of *OpenFlow*, the *OpenFlowRuleProvider* is responsible for generating *OFFlowMod* messages from **FlowRules**, and sending it to the appropriate network switch.

It relies on the *OpenFlow* subsystem for its message factories and *OpenFlowSwitch* references for this task.

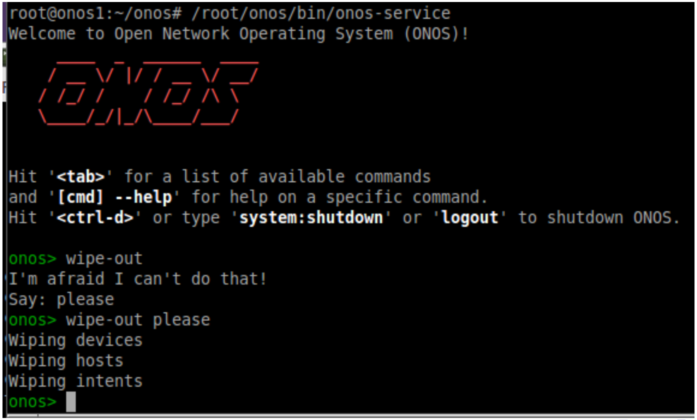
ONOS Intent Hands On

onos

Open Network Operating System

ONOS Intent Hands On Based On [Mar15]

$ docker run -td -p 8181:8181 -p 6633:6633 -h onos1 –name onos1 heitormotta/sci-onos:run
$ docker exec -ti onos1 bash
# ./bin/onos-service
onos>



```
root@onos1:~/onos# /root/onos/bin/onos-service
Welcome to Open Network Operating System (ONOS)!

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown ONOS.

onos> wipe-out
I'm afraid I can't do that!
Say: please
onos> wipe-out please
Wiping devices
Wiping hosts
Wiping intents
onos>
```

ONOS Intent Hands On Based On [Mar15]

first contact

onos> app activate org.onosproject.drivers (to be possible access http://localhost:8181/onos/ui/index.html)

mininet topology

$ sudo mn –topo tree,2 –controller remote,$(ipdocker onos1),6633 –mac

mininet> pingall (no connectivity)

openflow and proxyarp

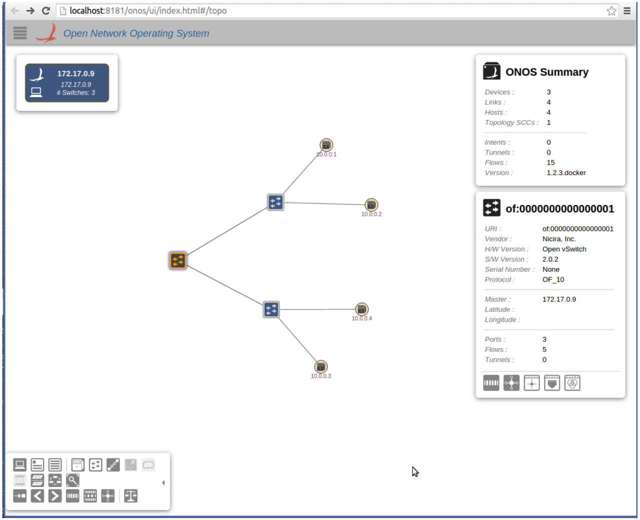
onos> app activate org.onosproject.openflow (to recognize devices)

onos> app activate org.onosproject.fwd

mininet

mininet> pingall (ok)
mininet> h1 ping h2

ONOS Intent Hands On Based On [Mar15]


localhost:8181/onos/ui/index.html#/topo
Open Network Operating System
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4

172.17.0.9
172.17.0.9
Switches: 3

ONOS Summary

Devices :	3
Links :	4
Hosts :	4
Topology SCCs :	1
Intents :	0
Tunnels :	0
Flows :	15
Version :	1.2.3.docker

of:0000000000000001

URI :	of:0000000000000001
Vendor :	Nicira, Inc.
H/W Version :	Open vSwitch
S/W Version :	2.0.2
Serial Number :	None
Protocol :	OF_10
Master :	172.17.0.9
Latitude :	
Longitude :	
Ports :	3
Flows :	5
Tunnels :	0

ONOS Intent Hands On Based On [Mar15]

simple-switchl2-app

$ onos-app $(ipdocker onos1) install target/switchl2-app-1.0-SNAPSHOT.oar

onos> app deactivate org.onosproject.fwd (ping stop to work)

onos> app activate simple.switchl2.app (ping work again)

intent-switch-app

onos> app deactivate simple.switchl2.app (ping stop to work)

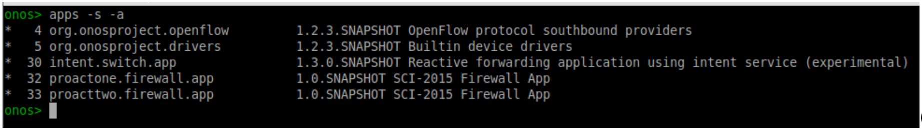
$ onos-app $(ipdocker onos1) reinstall! target/intent-switch-app-1.3.0-SNAPSHOT.oar

(ping work again)

proactive-firewall-app

$ onos-app $(ipdocker onos1) reinstall! target/proactive-firewall-1.0-SNAPSHOT.oar

ONOS Intent Hands On Based On [Mar15]

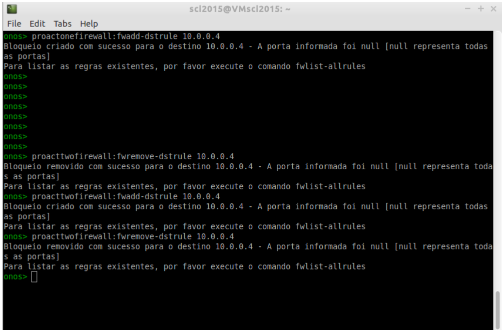


```
onos> apps -s -a
*   4 org.onosproject.openflow        1.2.3.SNAPSHOT OpenFlow protocol southbound providers
*   5 org.onosproject.drivers         1.2.3.SNAPSHOT Builtin device drivers
*  30 intent.switch.app               1.3.0.SNAPSHOT Reactive forwarding application using intent service (experimental)
*  32 proactone.firewall.app          1.0.SNAPSHOT SCI-2015 Firewall App
*  33 proacttwo.firewall.app          1.0.SNAPSHOT SCI-2015 Firewall App
onos>
```

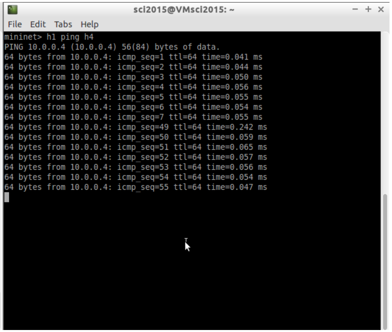
Cl

proactive-firewall in proactone-firewall and proacttwo-firewall

ONOS Intent Hands On Based On [Mar15]



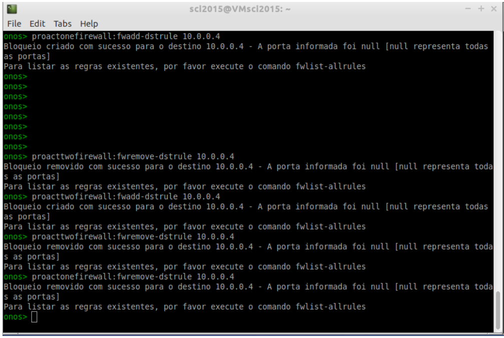
```
onos> proactonefirewall:fwadd-dstrule 10.0.0.4
Bloqueio criado com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa todas
as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos>
onos>
onos>
onos>
onos>
onos>
onos>
onos>
onos> proacttwofirewall:fwremove-dstrule 10.0.0.4
Bloqueio removido com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa toda
s as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos> proacttwofirewall:fwadd-dstrule 10.0.0.4
Bloqueio criado com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa todas
as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos> proacttwofirewall:fwremove-dstrule 10.0.0.4
Bloqueio removido com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa toda
s as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos>
```



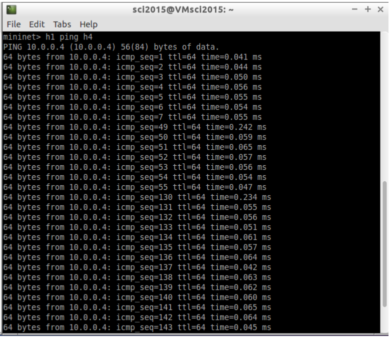
```
mininet> h1 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.055 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=0.054 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=0.055 ms
64 bytes from 10.0.0.4: icmp_seq=49 ttl=64 time=0.242 ms
64 bytes from 10.0.0.4: icmp_seq=50 ttl=64 time=0.059 ms
64 bytes from 10.0.0.4: icmp_seq=51 ttl=64 time=0.065 ms
64 bytes from 10.0.0.4: icmp_seq=52 ttl=64 time=0.057 ms
64 bytes from 10.0.0.4: icmp_seq=53 ttl=64 time=0.056 ms
64 bytes from 10.0.0.4: icmp_seq=54 ttl=64 time=0.054 ms
64 bytes from 10.0.0.4: icmp_seq=55 ttl=64 time=0.047 ms
```

proactonefirewall:fwadd-dstrule 10.0.0.4
proacttwofirewall:fwremove-dstrule 10.0.0.4 (fail)
proacttwofirewall:fwadd-dstrule 10.0.0.4
proacttwofirewall:fwremove-dstrule 10.0.0.4 (ok for few pings)

ONOS Intent Hands On Based On [Mar15]



```
onos> proactonefirewall:fwadd-dstrule 10.0.0.4
Bloqueio criado com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa todas
as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos>
onos>
onos>
onos>
onos>
onos>
onos>
onos>
onos> proacttwofirewall:fwremove-dstrule 10.0.0.4
Bloqueio removido com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa toda
s as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos> proacttwofirewall:fwadd-dstrule 10.0.0.4
Bloqueio criado com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa todas
as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos> proacttwofirewall:fwremove-dstrule 10.0.0.4
Bloqueio removido com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa toda
s as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos> proactonefirewall:fwremove-dstrule 10.0.0.4
Bloqueio removido com sucesso para o destino 10.0.0.4 - A porta informada foi null [null representa toda
s as portas]
Para listar as regras existentes, por favor execute o comando fwlist-allrules
onos> 
```



```
mininet> h1 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.055 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=0.054 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=0.055 ms
64 bytes from 10.0.0.4: icmp_seq=49 ttl=64 time=0.242 ms
64 bytes from 10.0.0.4: icmp_seq=50 ttl=64 time=0.059 ms
64 bytes from 10.0.0.4: icmp_seq=51 ttl=64 time=0.065 ms
64 bytes from 10.0.0.4: icmp_seq=52 ttl=64 time=0.057 ms
64 bytes from 10.0.0.4: icmp_seq=53 ttl=64 time=0.056 ms
64 bytes from 10.0.0.4: icmp_seq=54 ttl=64 time=0.054 ms
64 bytes from 10.0.0.4: icmp_seq=55 ttl=64 time=0.047 ms
64 bytes from 10.0.0.4: icmp_seq=130 ttl=64 time=0.234 ms
64 bytes from 10.0.0.4: icmp_seq=131 ttl=64 time=0.055 ms
64 bytes from 10.0.0.4: icmp_seq=132 ttl=64 time=0.056 ms
64 bytes from 10.0.0.4: icmp_seq=133 ttl=64 time=0.051 ms
64 bytes from 10.0.0.4: icmp_seq=134 ttl=64 time=0.061 ms
64 bytes from 10.0.0.4: icmp_seq=135 ttl=64 time=0.057 ms
64 bytes from 10.0.0.4: icmp_seq=136 ttl=64 time=0.064 ms
64 bytes from 10.0.0.4: icmp_seq=137 ttl=64 time=0.042 ms
64 bytes from 10.0.0.4: icmp_seq=138 ttl=64 time=0.063 ms
64 bytes from 10.0.0.4: icmp_seq=139 ttl=64 time=0.062 ms
64 bytes from 10.0.0.4: icmp_seq=140 ttl=64 time=0.060 ms
64 bytes from 10.0.0.4: icmp_seq=141 ttl=64 time=0.065 ms
64 bytes from 10.0.0.4: icmp_seq=142 ttl=64 time=0.064 ms
64 bytes from 10.0.0.4: icmp_seq=143 ttl=64 time=0.045 ms
```

proactonefirewall:fwremove-dstrule 10.0.0.4 (ok)

Conflict Resolution in SDN

- Over-view: Three Approaches

Conflict Resolution in SDN

- Over-view: Three Approaches
- Study Case: ONOS Intent SwitchL2

Conflict Resolution in SDN

- Over-view: Three Approaches
- Study Case: ONOS Intent SwitchL2
- Other Directions:

Conflict Resolution in SDN

- Over-view: Three Approaches
- Study Case: ONOS Intent SwitchL2
- Other Directions:
  - SDN Orchestration → Controller of Controllers [MVCM15]

Conflict Resolution in SDN

- Over-view: Three Approaches
- Study Case: ONOS Intent SwitchL2
- Other Directions:
  - SDN Orchestration $\rightarrow$ Controller of Controllers [MVCM15]
  - Multiple Access Communications $\rightarrow$ Entropy [CL15]

Conflict Resolution in SDN

- Over-view: Three Approaches
- Study Case: ONOS Intent SwitchL2
- Other Directions:
  - SDN Orchestration $\rightarrow$ Controller of Controllers [MVCM15]
  - Multiple Access Communications $\rightarrow$ Entropy [CL15]
  - BigData $\rightarrow$ Machine Learning [AWL15]

Conflict Resolution in SDN

- Over-view: Three Approaches
- Study Case: ONOS Intent SwitchL2
- Other Directions:
  - SDN Orchestration → Controller of Controllers [MVCM15]
  - Multiple Access Communications → Entropy [CL15]
  - BigData → Machine Learning [AWL15]
  - Problems → Protocols [MMKJA14]

Thanks! Questions?

References I

Ian F Akyildiz, Pu Wang, and Shih-Chun Lin, *Softair: A software defined networking architecture for 5g wireless systems*, Computer Networks (2015).

Jie Chuai and Victor OK Li, *A new upper bound on the control information required in multiple access communications*, Computer Communications (INFOCOM), 2015 IEEE Conference on, IEEE, 2015, pp. 1822–1830.

Andrew D Ferguson, Arjun Guha, Chen Liang, Rodrigo Fonseca, and Shriram Krishnamurthi, *Participatory networking: An api for application control of sdns*, ACM SIGCOMM Computer Communication Review, vol. 43, ACM, 2013, pp. 327–338.

References II

Luis Felipe Cunha Martins, *Curso de redes definidas por software*, Seminário de Capacitação e Inovação, RNP, 2015.

Ivan Marsa-Maestre, Mark Klein, Catholijn M Jonker, and Reyhan Aydoğan, *From problems to protocols: Towards a negotiation handbook*, Decision Support Systems **60** (2014), 39–54.

Raul Munoz, Ricard Vilalta, Ramon Casellas, and Ricardo Martinez, *Sdn orchestration and virtualization of heterogeneous multi-domain and multi-layer transport networks: The strauss approach*, Communications and Networking (BlackSeaCom), 2015 IEEE International Black Sea Conference on, IEEE, 2015, pp. 142–146.

OnLab, *ONOS Intent Framework*, `http://onos.wpengine.com/wp-content/uploads/2014/11/ONOS-Intent-Framework.pdf`, 2014, [Online; accessed 21-December-2015.

References III

Chaithan Prakash, Jeongkeun Lee, Yoshio Turner, Joon-Myung Kang, Aditya Akella, Sujata Banerjee, Charles Clark, Yadi Ma, Puneet Sharma, and Ying Zhang, *Pga: Using graphs to express and automatically reconcile network policies*, Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, ACM, 2015, pp. 29–42.