Enabling external routing logic in ONOS with Intent Monitor and Reroute service

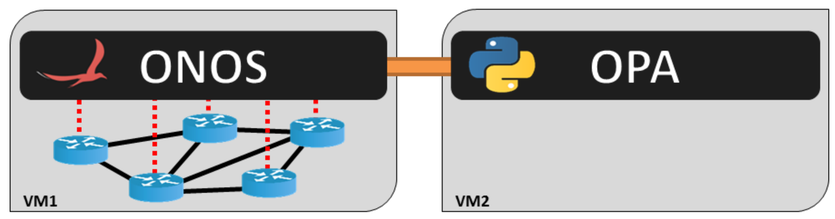
Davide Sanvito, Daniele Moro, Mattia Gullì , Ilario Filippini, Antonio Capone
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Italy

Andrea Campanella
Open Networking Foundation
Menlo Park, CA, USA

Abstract

SDN intent-based networking allows programmers to specify high-level policies without worrying about the low-level configurations which are compiled by the controller. Our Intent Monitor and Reroute service (IMR) [1] extends the capabilities of ONOS Intent Framework by allowing to both compile multiple intents together and to re-optimize their paths, according to the network state and their flow statistics, via off-platform applications (OPA). In this demo we show two examples of intent based ONOS application whose network performance can be improved, with no modifications to their code, exploiting the IMR service.

Demo setup



- 2 Ubuntu 16.04.3 (64 bit) server virtual machines (VM)
  - VM1 runs ONOS controller [2] and a Mininet [3] emulated network
  - VM2 runs then Off-Platform Application (in Python)
  - OPA and IMR service communicate via a REST API

ONOS Intent Framework

- Users specify high-level policies
- The SDN controller compiles them to low-level flow rules
  - shortest path
  - constraints on traversed devices/bandwidth
- Intents get re-compiled in case of network events
  - topology changes/failures

ONOS Intent Monitor and Reroute (IMR) service

- Extends ONOS Intent Framework's capabilities
- Intent monitoring enabled via IMR API or ONOS CLI
- IMR exposes a traffic matrix based on the intents to an Off-Platform Application (OPA)
- OPA implements external routing logic to suggest* new path:
  - Routing logic decoupled from the application
  - Reuse of existing TE tools
  - Joint compilation of multiple intents
  - Plug&Play external TE schemes (optimization tools/ML/AI)

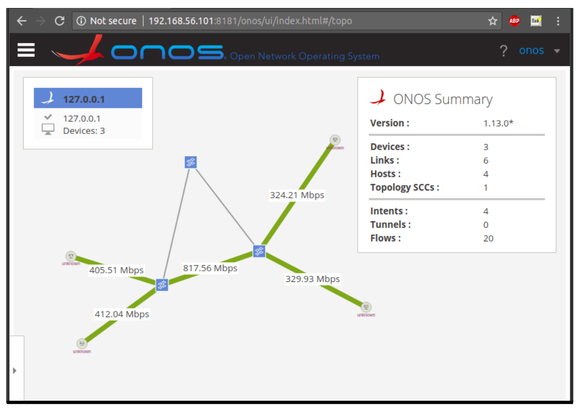
* the Intent Framework can still effectively recover from failures

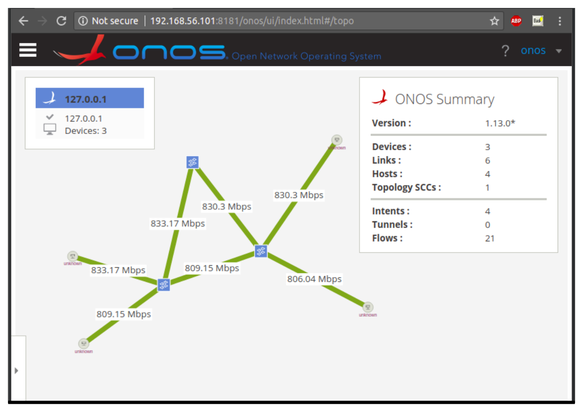
ONOS Intent Reactive Forwarding app

- ONOS sample app [4] reactively provides connectivity via intents
- Intents monitoring enabled via IMR service CLI command

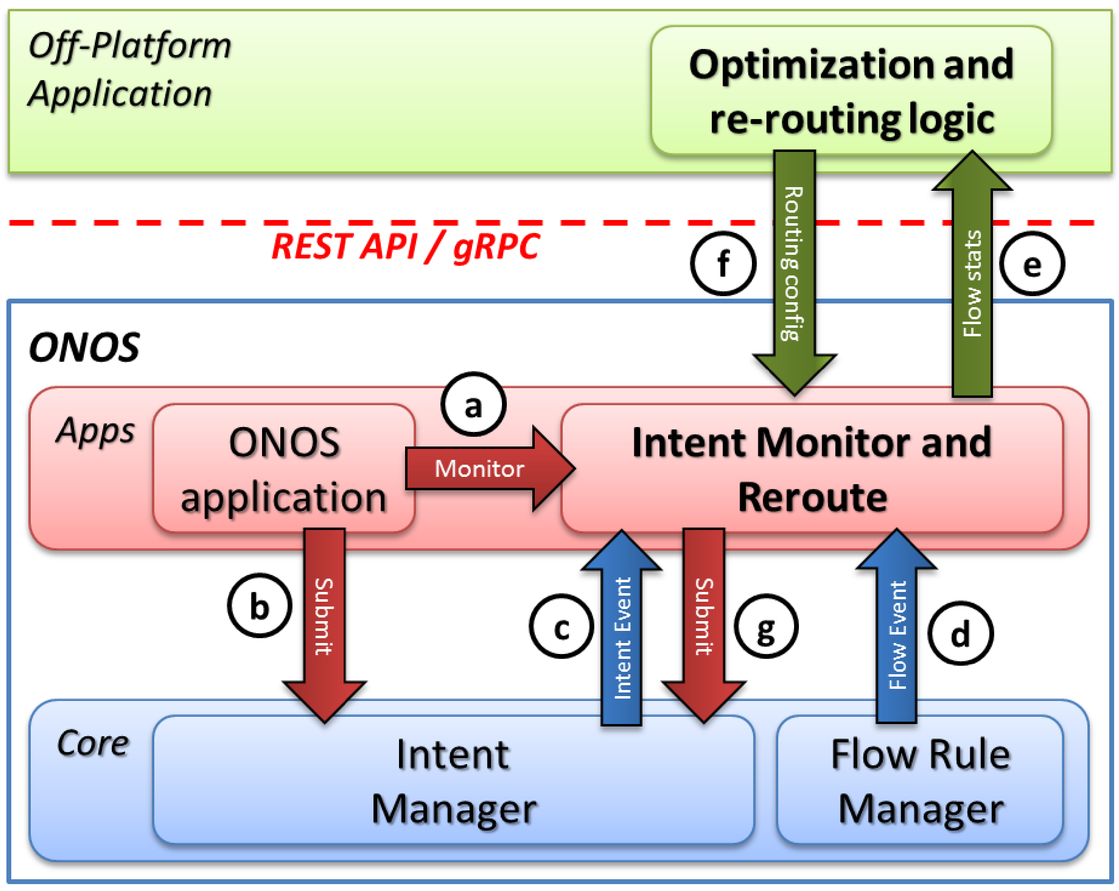
```
onos> imr:startmon appId appName [intentKey]
onos> imr:stopmon  appId appName [intentKey]
```

- No modification to application code
- OPA maximizes flows' throughput via greedy algorithm
- OPA code and tutorial available at ONOS Wiki [5]





ONOS, IMR service and OPA interactions



ONOS SDN-IP app

- ONOS sample app [6] provides AS-to-AS connectivity via intents according to BGP announcements
- Fine-grained intents monitoring via IMR service API

```
intentMonitorAndRerouteService.startMonitorIntent(intentKey);
```

- OPA iteratively runs the following
  - periodically collects TMs data from ONOS
  - solves the two optimization models [7] to minimize the MLU
  - schedules the activation of the robust routing configurations

References

[1] D. Sanvito, D. Moro, M. Gullì, I. Filippini, A. Capone, A. Campanella, "ONOS Intent Monitor and Reroute service: enabling plug&play routing logic" in IEEE NetSoft, 2018.
[2] P. Berde et al., "ONOS: Towards an Open, Distributed SDN OS," in ACM HotSDN, 2014
[3] B. Lantz, B. Heller, N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks" in ACM Hotnets, 2010.
[4] https://github.com/opennetworkinglab/onos-app-samples
[5] https://wiki.onosproject.org/display/ONOS/IMR+-+Intent+Monitor+and+Reroute+service
[6] https://wiki.onosproject.org/display/ONOS/SDN-IP
[7] D. Sanvito, I. Filippini, A. Capone, S. Paris, J. Leguay, "Adaptive Robust Traffic Engineering in Software Defined Networks" in IFIP Networking, 2018

POLITECNICO MILANO 1863