Demo Abstract: An Intent Solver for Enabling Intent-based SDN

Haodi Zhang, Yu Wang, Xiangtong Qi
Department of Industrial Engineering and Logistics Management
Hong Kong University of Science and Technology
Hong Kong, 999077
{howdy, ieemqi, ielmwangyu} @ust.hk

Weiping Xu, Tao Peng, Shucheng Liu
Department of R&D
Huawei Technologies Co., Ltd.
Shenzhen, China, 518000
{xuweiping, dr.pengtao, liushucheng} @huawei.com

***Abstract*—In recent years, software-defined networking (SDN) has been adopted in practice, such as the fast-growing market of software-defined wide area networks (SD-WAN). Intent, an emerging concept for SDN, aims to provide a simple interface for operators to manage the network. Under the framework of intent, an application submits a request by specifying the objective only, and the SDN controller decides how to reconfigure the network by considering the objectives of multiple intents with potential conflicts. For instance, in SD-WAN, when multiple applications or flows are competing for limited MPLS resource, we need to decide how and when to redirect some of them to Internet broadband connections. To this end, an intent solver is needed to analyze the received intents, making optimized solutions to operate the network. In this demo, we present the prototype of an intent solver focusing on intents related to flow routing. The demo takes intents both from applications for establishing/adjusting connections and from the network operator for load balancing, and generates optimized routing/rerouting solutions in real time.**

I. Introduction

The concept of intent has been proposed as an effective scheme to facilitate the operation of software-defined networks (SDN). Under of the principle of intent, an application only needs to submit some objective for accessing the network, without specifying the exact operations. The SDN controller will decide how to change the network configuration and resource allocation to satisfy the objective of the application. Major SDN platforms such as ONOS [1] and OpenDaylight [2] all provide frameworks to support intent.

Currently the concept of intent is still evolving, with a variety of understanding with respect to the scopes, features, and standards. For example, [3], [4], [5]. Sometimes people use other terms such as policy to refer to the similar concept.

While the idea of intent provides a promising scheme to simplify the network operation, there are challenges in its implementation. One difficulty lies in the fact that multiple applications will submit intents with inconsistent or even conflict objectives. The SDN controller has to detect and resolve any potential conflicts, which is a missing function of the current SDN controllers. To enable the idea of intent, some additional module is needed between the NBI and SDN controller. Such a module, which is referred to as Intent Solver in this demo, is responsible to analyze the objectives of different intents and generate an optimized solution to meet the needs. As far as we know, no similar work has been formally presented.

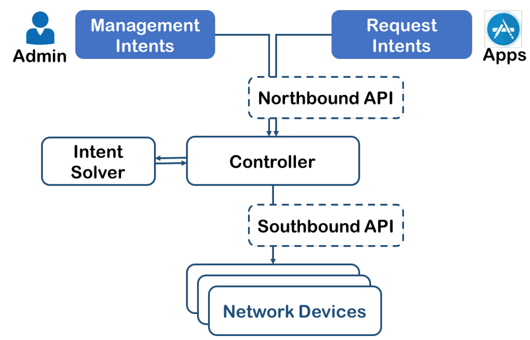


Fig. 1. System architecture

In this demonstration we present the design of an intent solver for SDN with heavy traffic, focusing on real-time optimization on flow routing. Each application sends intents to the SDN controller indicating the request for setting up or modifying a connection. Network administrator sends to the controller the so-called management intents, claiming the policies and requirements on managing the overall network. We use CERNET (The China Education and Research Network) as an example topology.

II. Intent and Intent Solver

Although the concept of intent can be applied to a broad range of situations for specifying diversified requests, we can roughly classify them into two categories, application intent and management intent. An application intent specifies the need of an individual application such as setting up the connection for a video conference or starting a virtual machine in cloud. A management intent is used by the network operator to regulate the status of the entire network. For instance, the network administrator may submit an intent as "each link in the network should not exceed 20% on utilization rate". The intent Solver needs to handle these two types of intents.

Figure 1 shows the architecture of our proposed intent-based SDN. The Intent Solver is a new module added to enhance the functionality of the SDN controller. The controller receives intents from network administrator and applications, passes the

978-1-5386-2784-6/17/$31.00 ©2017 IEEE

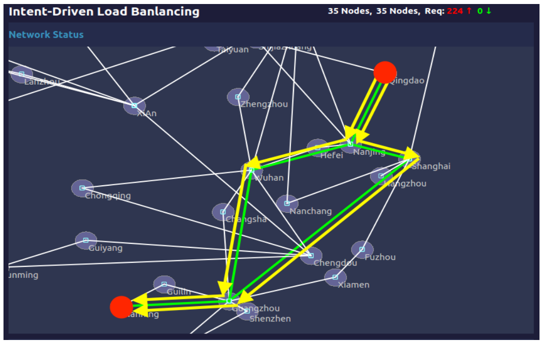


Fig. 2. Multiple routes

routing-related intents to Solver; then Solver generates optimized routing solution, returning it to the controller. We will demonstrate the Solver's capability by handling the following intents.

For applications: *build up/reconfigure/cancel a connection from source s to target t, with the expected bandwidth b.*

For the network administrator: *the utilization rate of each link should be below $m\%$.*

With the two categories of intents as input, the intent solver decides how to implement them. There are three modules in the solver.

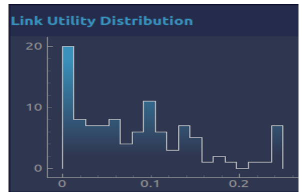
- **Modeller**: models the intent solving task into an optimization problem.
- **Optimizer**: computes the optimized solution.
- **Encoder**: encodes the solution from the optimizer into actions to take for controllers.

There are a few technical challenges in the Solver development. One is the time efficiency where a large scale multiple-commodity network flow problem needs to be solved quickly. Another one is the network stability where rerouting should be avoided or reduced to the least possible level, especially when the network administrator changes the network-wide policy or the network experience certain disruptions. We have designed tailor-made algorithms to overcome these challenges.

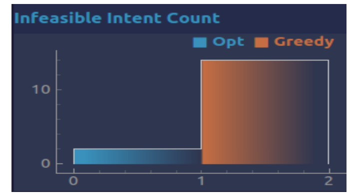
III. Demonstration

We develop a simulation package to demonstrate the effectiveness of the Solver. At any time, we generate a random number of intents from any pair of nodes on the network, with random bandwidths, and send them to the intent solver. The solver then generates the solution and the network status is updated. Some of the results are illustrated as follows.

- **Multiple routes.** We generate some request intent to increase the bandwidth of a connection from s to t. If the increase makes some link to reach or exceed the limitation $m\%$, the intent solver will direct the increased part to other routes from s to t. (Figure 2.)
- **Intent capacity.** We keep generating new request intents to install in the network. The flows in the network get randomly increase or decrease by intents. With a given utilization limitation $m\%$, we can get the distribution of the utilization rate of all links. (Figure 3 (a).) Compared with a greedy algorithm, the Solver can accommodate much more requests from application intents. (Figure 3 (b).)
  Figure 4 shows the numbers of rejected intents of the greedy algorithms and the Solver, when the network administrator changes the links' utilization rate limits.

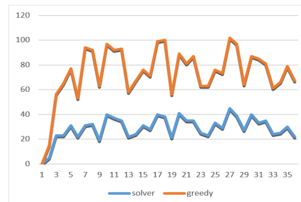


(a)

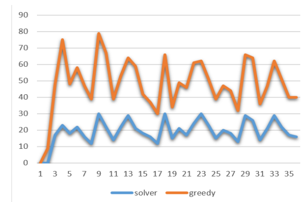


(b)

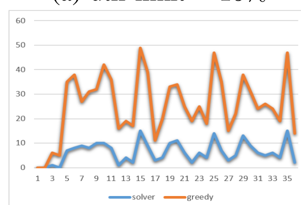
Fig. 3. (a) Distribution of utilization rate; (b) Numbers of rejected intents



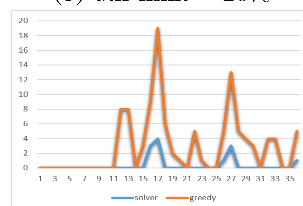
(a) util limit = 15%



(b) util limit = 20%



(c) util limit = 25%



(d) util limit = 30%

Fig. 4. Numbers of rejected intents with different utilization rates.

IV. Conclusion

In this demonstration, we present the design and implementation for an Intent Solver, a new module to enhance the functionality of SDN controller. The solver can successfully handle intents from multiple sources. We use flow routing related intents in this demonstration. The design can be extended to other types of intents, enabling the concept of Intent-based SDN, to be used in some specific scenarios or applications of SDN such as SD-WAN.

References

[1] "Open Network Operating System," 2014. [Online]. Available: http://onosproject.org

[2] "OpenDaylight Project," 2013. [Online]. Available: https://www.opendaylight.org

[3] Open Networking Foundation, "Intent NBI - definition and principles," *Theory and Practice of Logic Programming*, 2016.

[4] C. Prakash, J. Lee, Y. Turner, J. Kang, A. Akella, S. Banerjee, C. Clark, Y. Ma, P. Sharma, and Y. Zhang, "PGA: using graphs to express and automatically reconcile network policies," in *SIGCOMM 2015*. ACM, 2015, pp. 29–42.

[5] S. Paris, A. Destounis, L. Maggi, G. S. Paschos, and J. Leguay, "Controlling flow reconfigurations in SDN," in *INFOCOM 2016*. IEEE, 2016, pp. 1–9.