Magellan: Toward a Unified SDC Programming Framework with Automatic, Fast Updates

Kai Gao (Yale), **Geng Li** (Yale), **Taishi Nojima** (Yale), **Sanat Khurana** (Yale), **Chris Leet** (Yale), **Patrick Baker** (Dstl), **Franck Le** (IBM US), **Yeon-sup Lim** (IBM US), **Vinod K. Mishra** (ARL), **Y. Richard Yang** (Yale).

Magellan Framework

We develop a high-level, unified SDC programming framework which achieves unified control of both networking (i.e., SDN) and computing (i.e., network functions) resources, under the highly dynamic environment of SDC.

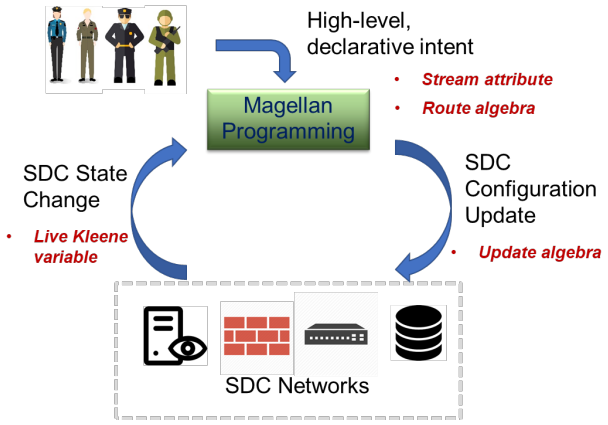

Figure 1: Magellan SDC programming framework.

Key components of the framework:

- **Stream attribute**: a simple abstraction for modeling cross-packet states extracted by SDC network functions;
- **Route algebra**: a simple yet powerful abstraction to flexibly express consistent, advanced routing;
- **Live Kleene variable**: a unified data model and the foundation of Magellan, which enables built-in support for unknown and asynchronous data, as well as semantic dependency tracking and automatic incremental updates.
- **Update algebra**: a systematic, theoretical framework based on abstract algebra, to enable continuous, nonblocking, fast composition of multiple updates.

Demo Description

We consider a simple network shown in Figure 2, in which an SDC state monitor is deployed to monitor the mission state from the three soldiers. Whenever a soldier reports an alert, the mission state becomes alert.

Mission-aware routing:

- **Alert state**: A can only download data using path A-B-C-D
- **Non-alert state**: A prefers path A-E-C-D to A-B-C-D for better bandwidth

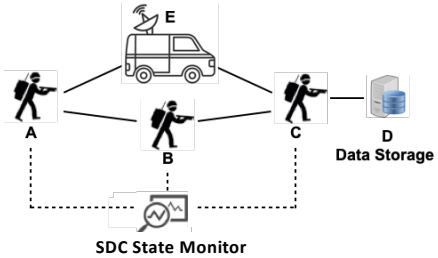

Figure 2: A simple network.

Demo Program

We use the following simple SDC program and two coherent scenarios to demonstrate the functions and benefits of Magellan unified SDC control with automatic update.

```
1  //magellan.example.DemoProgram
2  object DemoProgram extends SDCProgram {
3    import magellan.example.SPC.global_state
4
5    val G = NetworkTopology("topology")
6
7    val X = SimplePath(G, A :—: B :—: C :—: D)
8    val Y = SimplePath(G, A :—: E :—: C :—: D)
9
10   override def onPacket(pkt: Packet) = program {
11     iff (pkt.global_state === "on_alert") {
12       val Z = any(Y >> X)
13       bind(pkt, Z)
14       bind(inv(pkt), inv(Z))
15     } else {
16       bind(pkt, any(X))
17       bind(inv(pkt), inv(X))
18     }
19   }
20 }
```

Figure 3: A high-level SDC program illustration example selecting paths for packets based on the global_state information of packets.

Scenario 1

Programming with Live Variables for Automatic Tracking and Updates Supporting Unified, Asynchronous Programming of Networking and SDC Network Functions

- **Mission-aware routing**: Soldiers updates new mission states, and the program can generate new routes automatically
- **Link failure**: A new route will be computed automatically and the update will be deployed efficiently.

Scenario 2

Update Algebra for Fast, non-blocking Update Distribution

- **Frequent mission state changes**: Updates will be composed to reduce redundancy.

Publication(s) & Impact

- Kai Gao, Taishi Nojima, and Y. Richard Yang. 2018. Trident: toward a unified SDN programming framework with automatic updates. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (SIGCOMM '18), pp. 386-401 ACM, New York, NY, USA,.
- Geng Li, Y. Richard Yang, Franck Le, Yeon-sup Lim, and Junqi Wang. "Update Algebra: Toward Continuous, Non-Blocking Composition of Network Updates in SDN." In *2019-IEEE Conference on Computer Communications* (INFOCOM '19), pp. 1081-1089. IEEE, 2019.