Flipper: Fault-Tolerant Distributed Network Management and Control

Subhrendu Chattopadhyay,
Niladri Sett, Sukumar Nandi, and Sandip Chakraborty

May 8, 2017

Content

Example Scenario

- An academic institute Just like IIT, Guwahati
- Sys admin wants to distribute bandwidth policies based on network usage
- Not scalable
- Minor misconfiguration may lead to network underutilization

Problems of Traditional Architecture

- Lack of programmability
- Complex architecture
- Customized protocols for heterogeneous hardware platform and vendor dependence
- Delay in deployment
- Resource management and inconsistent policies.

Definition

- **Data and control plane separation**
- **Controller based decision**
- **Flow based decision**
- **Programmable network**

SDN with distributed controller

- Required for improved scalability
- e.g ONIX[1], ONOS[2]
- ONIX uses two types of data bases
  - Transactional database for high level network rules.
  - DHT-based database for volatile network state information.
- Controller Placement trade-off: Number of controller vs control plane overhead[3],

[1] Teemu Koponen et al. "Onix: A Distributed Control Platform for Large-scale Production Networks". In: *Proceedings of the 9th USENIX Conference on OSDI, 2010.* USENIX Association, 2010, pp. 1–6.

[2] Pankaj Berde et al. "ONOS: towards an open, distributed SDN OS". . In: *Proceedings of the 3rd HotSDN, 2014.* ACM. 2014, pp. 1–6.

[3] Soheil Hassas Yeganeh, Amin Tootoonchian, and Yashar Ganjali. "On scalability of software-defined networking". In: *IEEE Communications Magazine*, 51.2 (2013), pp. 136–141.

SDN with distributed controller

- POCO-PLC[4]
  - Off-line placement of controllers.
  - Fault-resilience towards node or double link failure.
  - Claims 20% of needs nodes needs to be deployed as controller for most practical small scale topology.

[4] David Hock et al. "POCO-PLC: Enabling Dynamic Pareto-Optimal Resilient Controller Placement in SDN Networks". In: *Proceedings of the 33rd INFOCOM, 2014* (2014).

Issues with POCO-PLC and SDN

- POCO-PLC
  - Requires SDN enabled infrastructure
  - Does not cope up with arbitrary link/node failure.
  - Off-line solution

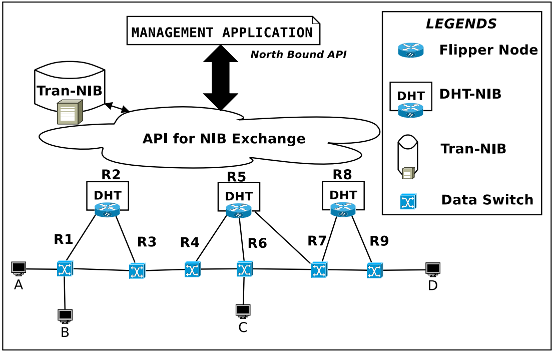
Proposal: Flipper Architecture

- COTS devices acts as PDEP.
- Uses NFV to achieve this feature[5]
  - Based on ONIX, tran-NIB and DHT-NIB.
  - Each nodes are called flipper.
  - Each flipper can act as either DHT-NIB or switch.
- DHT-flipper can convert itselves to switch flipper dynamically (and vice versa)

[5] M Said Seddiki et al. "Flowqos: Qos for the rest of us". In: *Proceedings of the 3rd HotSDN, 2014*. ACM. 2014, pp. 207–208.

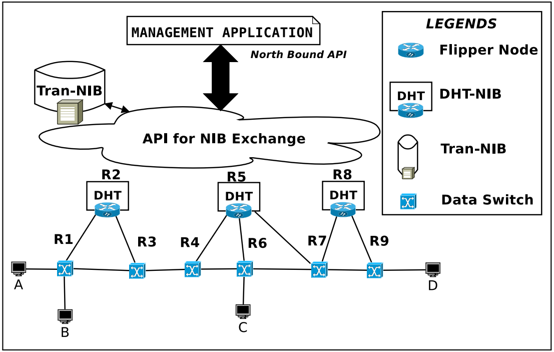
Proposal: How Flipper works

- DHT-flipper: Hosts: A,B,C,D
- tran-NIB: High level network rules (e.g ACLs etc.)
- Switch-flipper: Acts as forwarding device



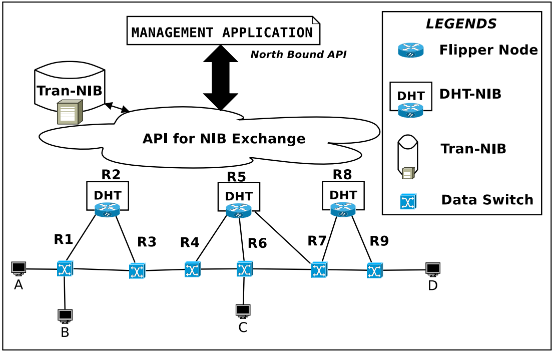
Proposal: How Flipper works

- DHT-flipper: Acts as NIB for volatile network information. (e.g. Link statistics)
- DHT-flipper requires to be placed within one-hop of distance of the switch.



Flipper: Failure Use-Case

- R5 fails.
  - R4 and R6 can detect failure.
- R4, R6 readjusts new locations of DHT-NIBs.

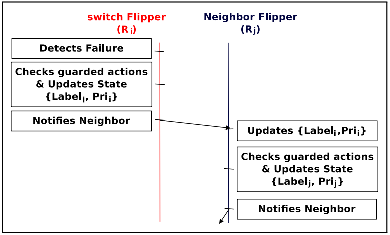


Fault-tolerant Flipper Readjustment

- Algorithm is represented as Guarded statements.

  $(Ruleno)| < Guard > \rightarrow < Action >$

- Each guarded statement execution timing diagram is given in the figure.



Fault-tolerant Flipper Readjustment

Variables:

$Label_i = \{NIB, Swi, Wait\}$

$Pri_i = \{0, 1, \ldots, B\}$

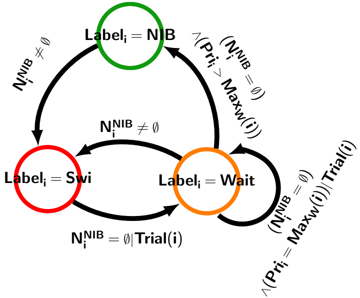
Functions:

$N^{NIB}(i) = \forall j \in N_i : Label_j = NIB$

$N^{Wait}(i) = \forall j \in N_i : Label_j = Wait$

$Max_W(i) = \forall j \in N_{Wait} : Max(Pri_j)$

$Trial(i) Pri_i = Rand(0, 1, \ldots, B)$



Properties of Flipper Readjustment

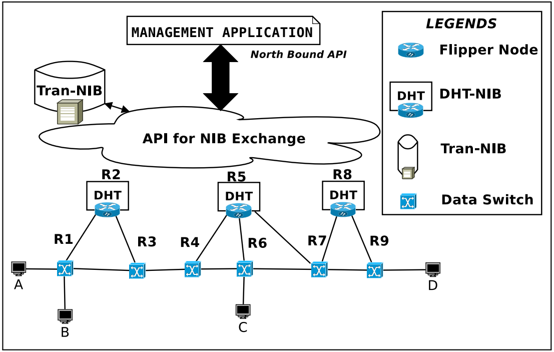
- If any flipper in the system is in intermediate state then there is at least one rule which can be executed further.
- If the system is in a state where flippers with DHT-flippers form a MIS, it will remain in that state forever, provided no further fault occurs. (Closure property)

Properties of Flipper Readjustment

- If X denote the random variable indicating the number of rounds required to find a unique maximum priority in the closed neighborhood of v then $E[X] \leq e$, where e represents Euler-Mascheroni constant.
- The expected number of moves for convergence is $O(n)$.

Properties of Flipper Readjustment

- Flipper is partition tolerant:
  - Say, R3 and R4 fails.
  - In such cases the R1 and R3 invokes the flipper readjustment.
  - A new DHT-flipper is chosen in their vicinity.



Simulation Results

- Based on NS3.
- Comparison with POCO-PLC
- 3 different topologies are used.
  - Synthetic Grid (64x64 nodes)
  - AS733 real dataset[6]
  - Oregon real dataset[7]

[6] *SNAP Autonomous systems AS-733 data set*. http://snap.stanford.edu/data/as.html.

[7] *SNAP Autonomous systems - Oregon-1 data set*. http://snap.stanford.edu/data/oregon1.html.

Simulation Result

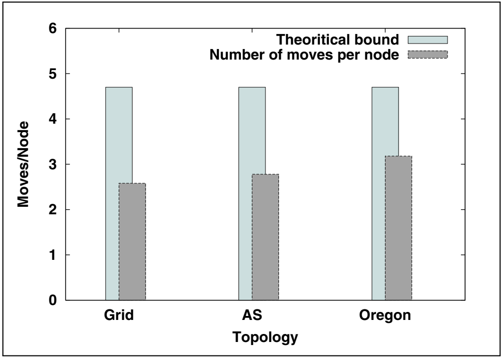


Figure : Number of moves executed per node

Simulation Result

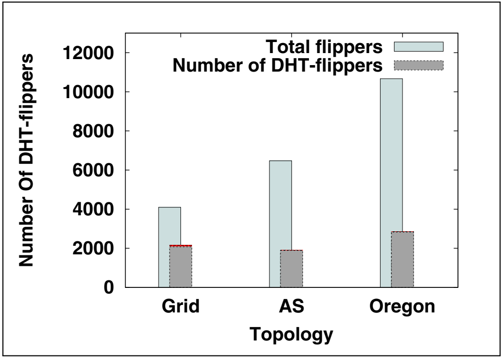


Figure : Number of placed controllers

Simulation Result

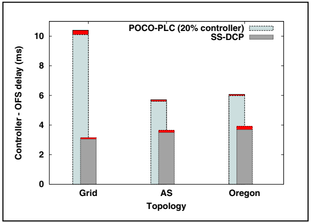


Figure : Number of moves executed per node

Summery of Simulation Results

- Number of required Flipper depends on the topology.
- 5% 10% increase in number of DHT-flipper can reduce flow setup delay by more than 60% for both of the real networks.
- The performance improvement in terms of flow initiation delay is due to the fact that, each switch-flipper has a DHT-flipper in its neighborhood.

Emulation Results

- 50 node topology taken from Oregon dataset.
- 200 random flows
- Mininet for emulation.
  - Experiment 1: The selected flippers are 1-hop away from each other.
  - Experiment 2: The selected flippers are more than 2 hops distance apart.

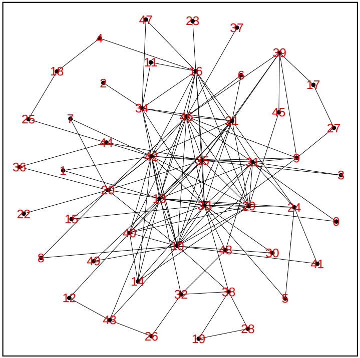


Figure : Used Topology

Emulation Results

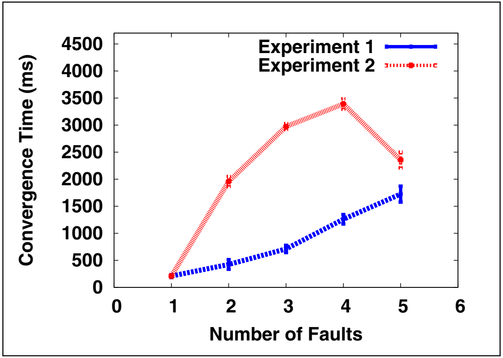


Figure : Convergence time vs number of flipper failure

Emulation Results

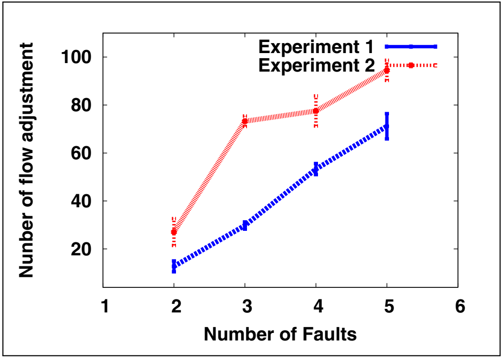


Figure : Number of flow adjustment readjustment vs number of flipper failure

Emulation Results

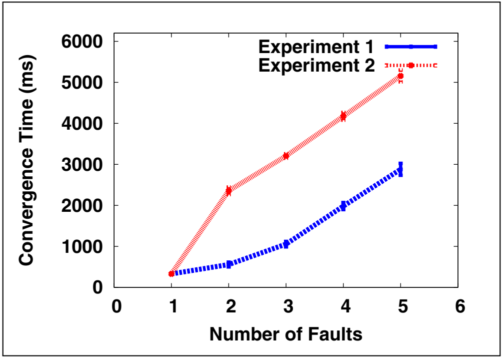


Figure : Convergence time vs number of link failure

Emulation Results

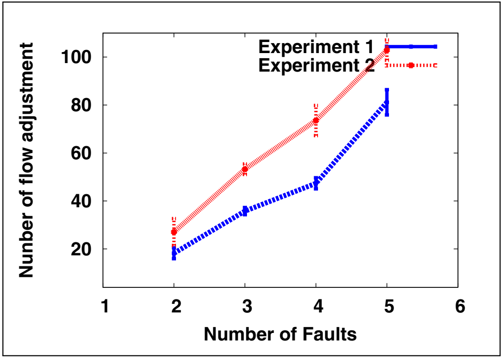


Figure : Number of flow adjustment readjustment vs number of flipper failure

Summery of Emulation Results

- Convergence time is dependent on the separation of the failed flippers or failed links.
- Increase in number of flipper failure or link failure increases the number of flows required to be rerouted.
- The performance improvement in terms of flow initiation delay is due to the fact that, each switch-flipper has a DHT-flipper in its neighborhood.

Future Plan

Flipper:

- Supports SDN like network management and control.
- Avoids the controller bottleneck problem.
- Supports a stronger notion of fault tolerance.
- Provides a scalable notion of dynamic role adaptation.

Thank You