Medieval: Towards A Self-Stabilizing, Plug & Play, In-Band SDN Control Network

Liron Schiff Stefan Schmid Marco Canini

TEL AVIV UNIVERSITY

Telekom Innovation Laboratories

UCL Université catholique de Louvain

Main Contributions

Medieval: A Plug & Play Distributed SDN Control Plane

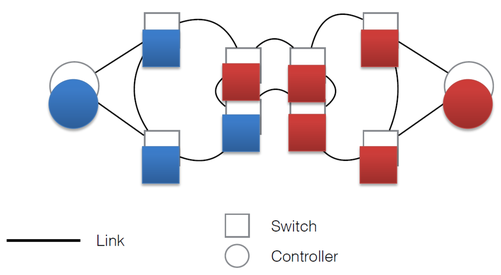
- Flexible controller membership (additions, removals, failures)
- Automatic topology, controller and switch discovery & management
- Supports ONIX, ElastiCon, Beehive, STN, and more.

Provable Self-Stabilization

- *Self-Stabilization*: Important concept in fault-tolerant distributed systems: Converge to "good state" from an arbitrary initial state.
- Medieval tolerates failures and delays: low re-convergence times

Background + Model

Network Managed by Multiple Controllers



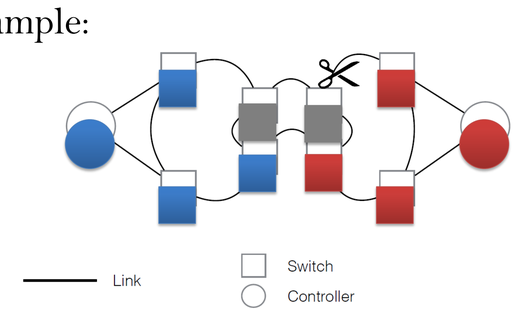
"Good network state" :=

- Every switch is connected to a controller.
- Controllers can communicate and make joint decisions.

Why in-band? No reason to build, operate, and ensure the reliability of a separate out-of-band network. Also, out-of-band networks are typically underprovisioned and have limited redundancy.

Failure Resiliently

- Example:



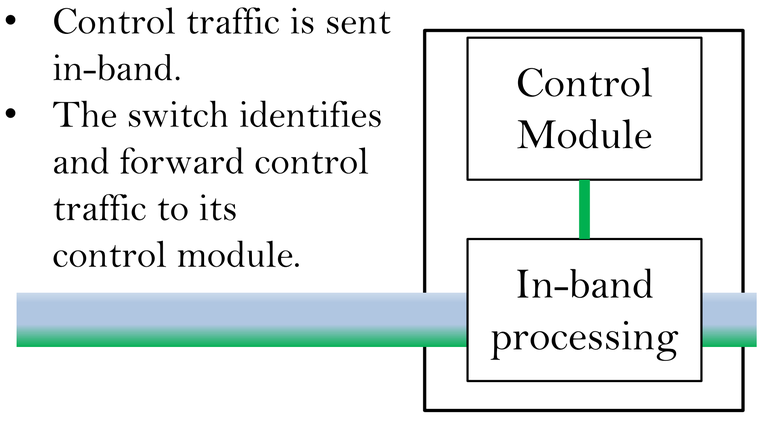
- Goal: Network should return to a good state.

The Medieval Approach

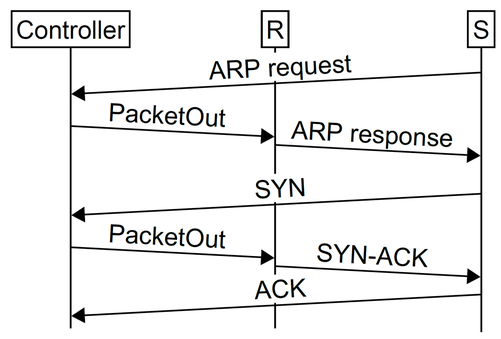
- Controllers aim to continuously grow their management regions...
- … and "conquer" unmanaged switches.
- Management with two spanning trees:
  (1) Per-region spanning tree (bidirectional, owned by controller)
  (2) Network-wide spanning tree (to connect controllers)

Switch Structure

- Control traffic is sent in-band.
- The switch identifies and forward control traffic to its control module.

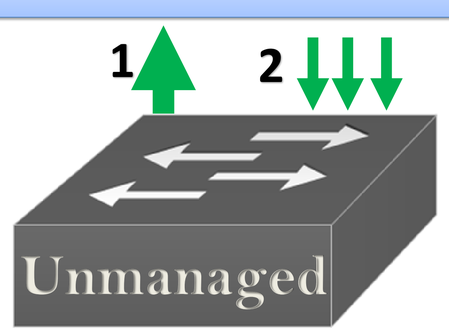


The Protocol



Controller uses a managed switch, R, to detect and establish connection to a new switch S.

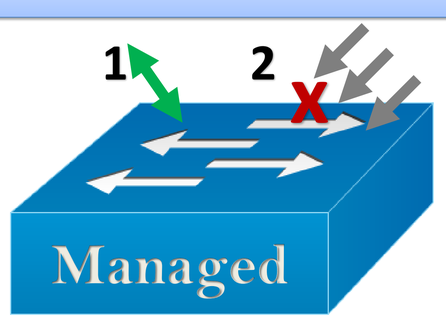
Switch States



1. Broadcast
2. Any controller can respond

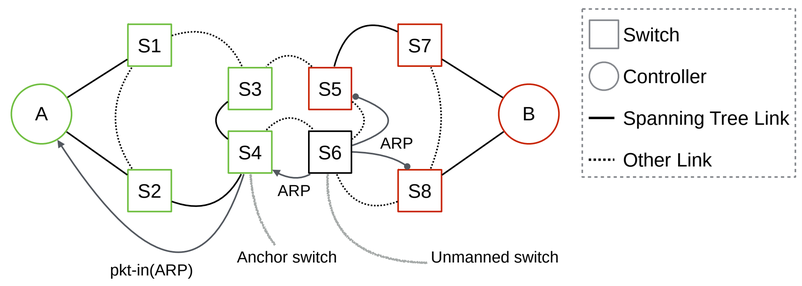
Session established

No keep-alive timeout



1. Controller traffic is passed through
2. Other controllers are blocked

Controller to Switch Connectivity



Controllers "conquer" switches adjacent to their regions of control and build a spanning tree for controller-to-switch connectivity.

Evaluation

Prototype Implementation

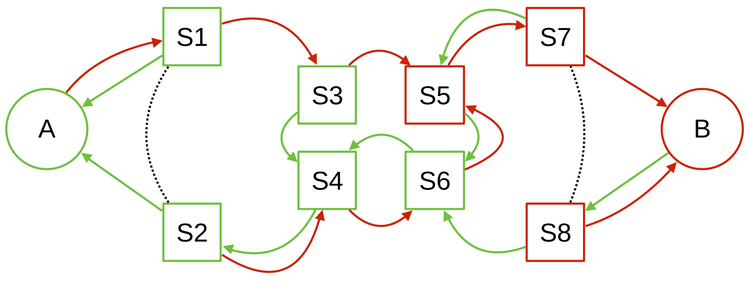
- Emulator in Java
- OpenFlow switches and controllers: light-weight threads
- Links modelled by message queues
- Fat-tree topology (k=4), 1-8 controllers
- Measured time to manage switches

Formal Analysis

Theorem 1. Medieval is self-stabilizing: Given any initial configuration and set of controllers, Medieval will establish a communication network between controllers and any physically connected network.

# ctrls	1	2	3	4	5	6	7	8
Time (ms)	9382	6983	6150	4224	6035	5104	3704	3680

Controller to Controller Connectivity



Per-controller global spanning trees provide controller-to-controller connectivity.

Related Work

1. T. Koponen et al. Onix: A Distributed Control Platform for Large-scale Production Networks. In OSDI, 2010.
2. A. Dixit et al. Towards an Elastic Distributed SDN Controller. In HotSDN, 2013
3. S. H. Yeganeh et al. Beehive: Towards a simple abstraction for scalable software-defined networking. In HotNets, 2014.
4. M. Canini et al. A Distributed and Robust SDN Control Plane for Transactional Network Updates. In INFOCOM, 2015.