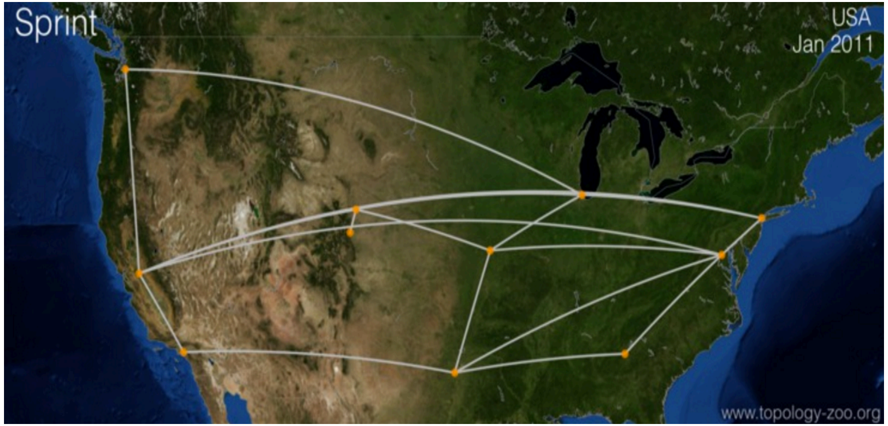
RELIABILITY IN SDN CONTROL PLANE

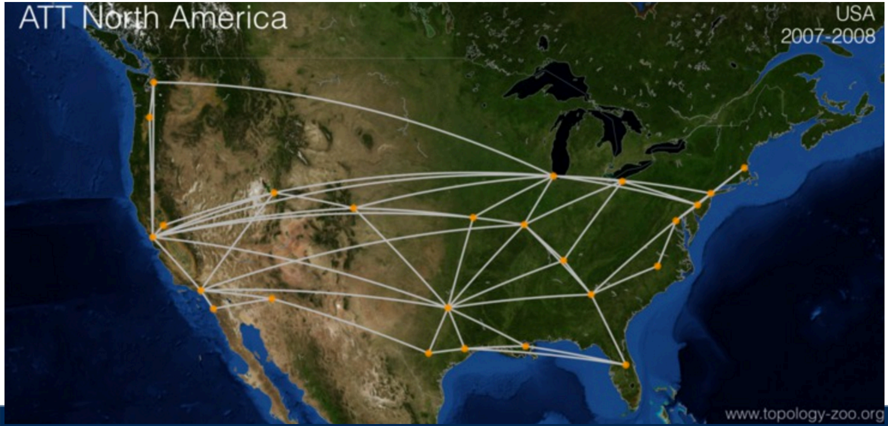
Sedef Savas

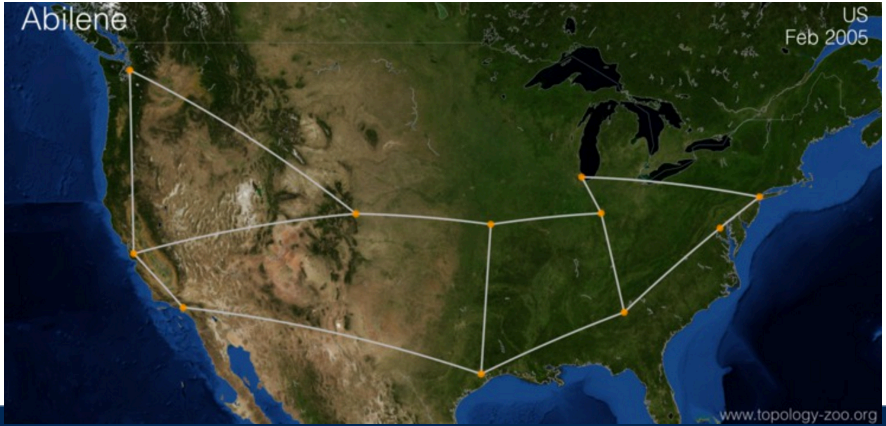
June 30, 2017

Netlab Friday Group Meeting

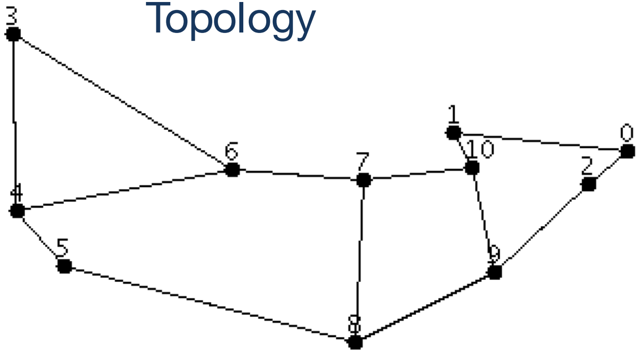
Networks from topology-zoo.org



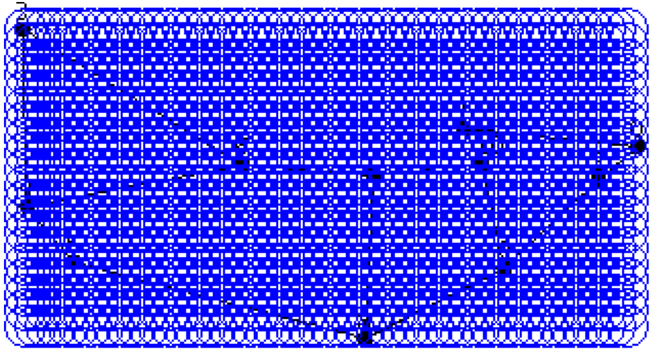




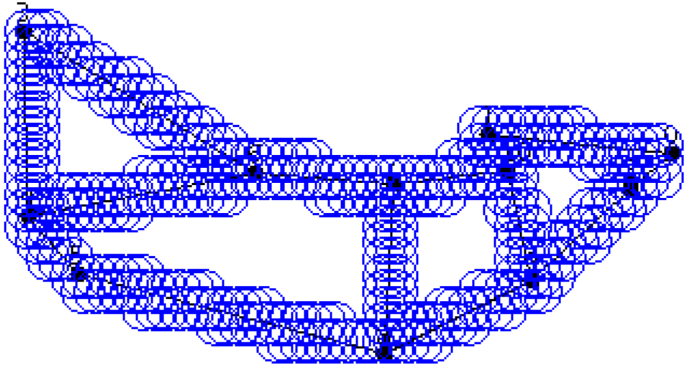
Topology



Moving 100 km radius disaster: #1080

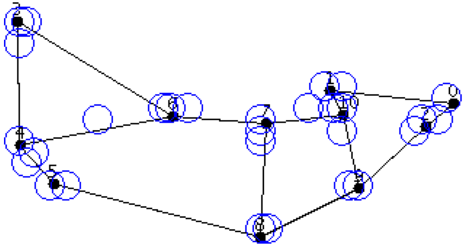


Disasters affecting > 0 element: #302

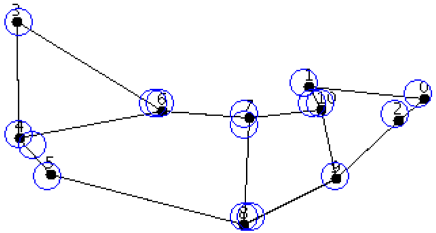


Distinct disasters

affect > 0 element: #29

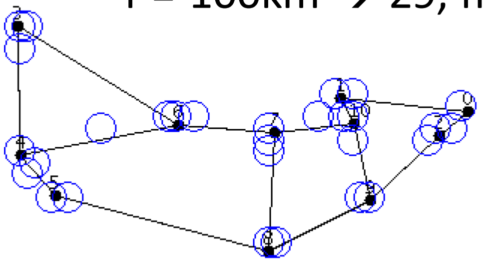


affect > 1 element: #16

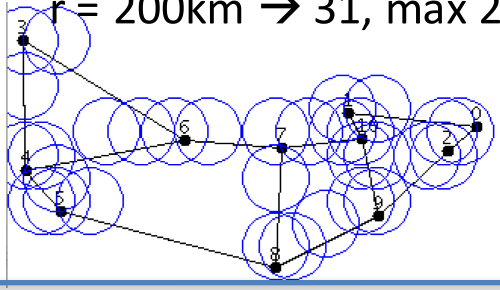


Affect of Disaster Radius on # of Disasters

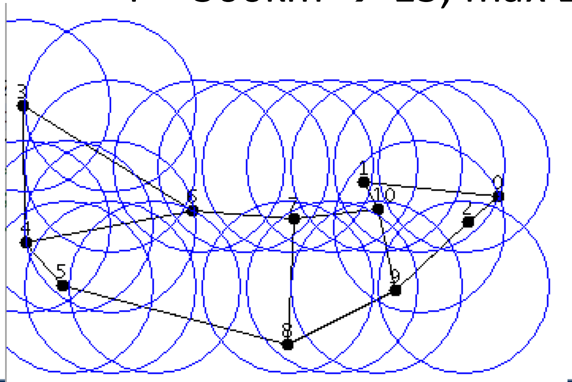
r = 100km → 29, max 1 node down



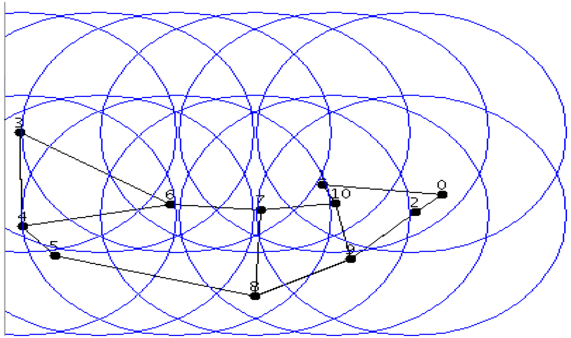
r = 200km → 31, max 2 nodes down



r = 500km → 15, max 2 nodes down



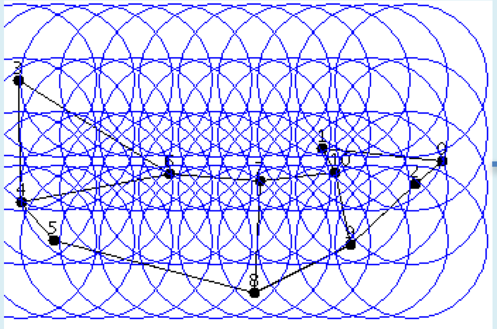
r = 1000km → 12 max 5 nodes are down



UCDAVIS

Affect of Disaster Radius on # of Disasters

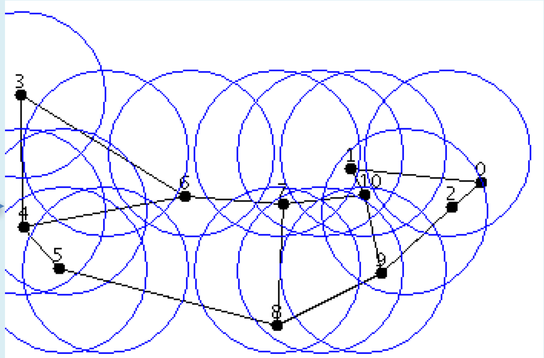
r = 500km → 44 disaster circles,
max 2 nodes down



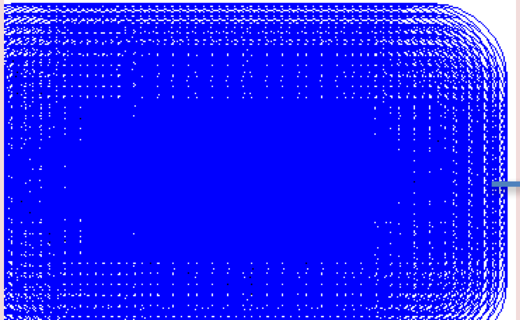
Eliminate same-effect disasters
and disasters affect < 2 links

We miss some combinations.

15 out of 44 affects > 1 element

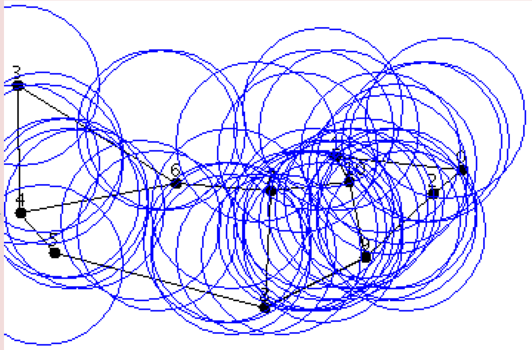


r = 500, walk = 100 → 1080 disaster circles,
max 3 nodes down



Eliminate same-effect disasters
and disasters affect < 2 links

46 out of 1080 affects > 1 element



RTTs don’t vary substantially.

Monthly Network Averages	Target Values	Observed Values May	April	March
U.S. Network Averages				
Roundtrip Latency	< 37 ms	32.6	32.9	33.1
Roundtrip Loss*	< 0.05%	0.00%	0.00%	0.01%
Network Reliability	> 99.95%	99.9991%	99.9999%	99.9997%
Modem Connect Success Rate	> 95%	99.99%	99.99%	99.99%
Network Jitter	< 1 ms	0.57	0.58	0.58

at&t

SF to NYC is 4500km → 44 ms only transmission RTT

U.S. Network Latency

Figures are in **ms**. Thresholds are distance sensitive.

CITY PAIRS	Atl	Aus	Cam	Chi	Cle	Dal	Den	Det	Hou	Ind	Kan	LA	Mad	Nas	NO	NY	Orl	Pa	Phx	SA	SD	SF	StL	Sea
Austin	22																							
Cambridge	28	49																						
Chicago	24	27	23																					
Cleveland	18	35	18	7																				
Dallas	20	6	49	28	29																			
Denver	35	24	44	20	27	19																		
Detroit	22	38	22	7	4	34	26																	
Houston	19	7	50	32	34	6	25	39																
Indianapolis				5																				
Kansas City	23	15	32	12	18	10	15	19	15															
Los Angeles	51	34	70	46	52	32	27	52	36		41													
Madison				5							18													
Nashville	6	24		18	11	18	32	15	24		17	49												
New Orleans	12	13		36	33	13	31	36	7		22	43		16										
New York	23	45	6	19	13	40	37	22	41		27	64		25	35									
Orlando	10	28	37	33	27	27	45	32	22		32	58			14	34								
Philadelphia	21	46	8	18	11	39		14	41		25	63		21	32	3	32							
Phoenix	41	21	65	44	47	19	37	53	24		29	11		39	31	62	45							
San Antonio	22	3	54	35	37	7	26	40	5		17	30		26	12	46	26	47	20					
San Diego	43	29	73	49	56	27	30	55	31		37	4		45	38	67	53	67	8	26				
San Francisco	58	45	70	47	51	41	30	54	44		44	9		61	51	67	65	61	20	39	13			
St. Louis	19	25	27	7	14	20	21	14	25	13	6	49	11	12	32	22	27	20	39	26	44	50		
Seattle	71	58	70	47	53	52	34	53	58		49	27		64	64	65	78	63	36	59	29	16	53	
Washington	19	41	11	22	14	36	42	23	44		22	67		25	30	5	29	4	55	41	65	66	17	68

Current Overall Average: **32 ms**

We can assume overall RTTs are 1.5 x transmission delays

International Path Latency

Trans-Atlantic Path		
Wash	91	Frankfurt
Trans-Pacific Path		
SF	141	Hong Kong

Legend | Increasing Latency

NOTE: Please refresh this page every 15 minutes to ensure that the most current data is displayed.

A single controller can not keep flow setup time consistent or within acceptable limits, which is reported to be **200ms** in for mesh restoration.
[Boutaba et. al, “Dynamic Controller Provisioning in SDN” , 13]

http://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html

Flow Setup Latency

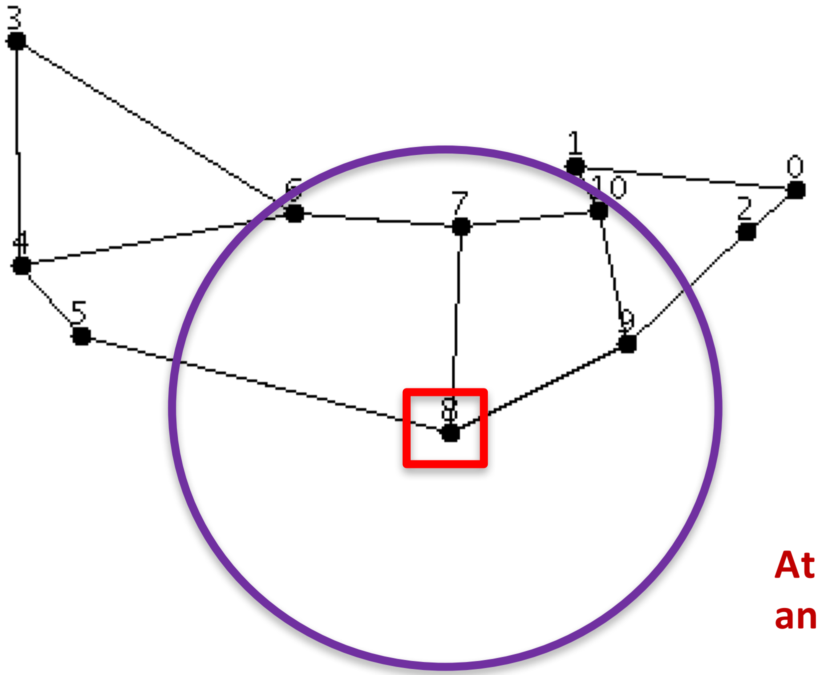
[*] Setting up routes in **cellular networks** (when a device becomes active, or during handoff) must complete within 40ms to ensure users can interact with Web services timely.

Flow setup time = speed of control programs, and latency to/from controller
+ switch modifying forwarding state as dictated

Inbound latency is involved in switch generating events (e.g., when a flow is seen for the first time) can be high (8 ms per rule).

Outbound latency is involved in the switch installing/modifying/deleting forwarding rules provided by control applications, is high as well (3ms - 30ms per rule for insertion and modification).

 [*] Mazu: Taming Latency in Software Defined Networks, Bell Labs, 2011.



Create Reachability Circles for Selecting Controllers

Node 8: 1, 6, 7, 8, 9, 10

Node 9: 0, 1, 2, 7, 8, 9, 10

•
•
•

At least 2 closeby controllers will be up for each swich, and at least 1 of them will survive after any possible disaster

Create disaster-joint sets for reachability circles:

Node 8: {1, 6, 7}, {8, 9, 10}

Node 9: {0, 1}, {2, 7}, {8, 9}, 10

Less than 2 disaster-joint sets or less than 2 controllers makes solution infeasible.
So, low S-C latency cause infeasible solutions.

Feasibility Score of DC Nodes

Prioritize **more reachable (in terms of latency), less disaster-prone,** and **more connected** DC nodes to place controllers.

$$\textbf{Feasibility score} = (\boldsymbol{\alpha} * \boldsymbol{R} + \boldsymbol{\beta} * \boldsymbol{D} + \boldsymbol{\gamma} * \boldsymbol{C})$$

R = Rank of nodes based on reachability (more reachable nodes are favorable.)

D = Rank of nodes based on # of disasters that affects them.

C = Edge-connectivity of nodes.

Based on this score, initial # and location of controllers are determined.

Switch-Controller Assignment

Least-reachable-selected controllers will be assigned first.

Set a max value for number of switch per controller.

How to Guarantee Enough Capacity After a Disaster?

Load-balancing is important but increase # of controllers.

CONTROL PATH MANAGEMENT FRAMEWORK FOR ENHANCING SDN RELIABILITY

IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT,
VOL. 14, NO. 2, JUNE 2017

Sejun Song[1], Hyungbae Park[1], Baek-Young Choi[1], Taesang Choi, and Henry Zhu[2], University of Missouri–Kansas City[1], Cisco[2]

Introduction

In SDN due to new and multi-lateral network domains, **critical challenges to achieve the same reliability services as existing networks.**

Control path network lies between a control and a data plane to connect them through in-band SDN or an out-of-band traditional network.

They propose a **control path management framework** to enhance SDN reliability.

New Reliability Challenges

- Traditional networks use distributed reliability protocols (heartbeat mechanisms, no heartbeat = failure).
- SDN creates a new network plane between control/data planes.

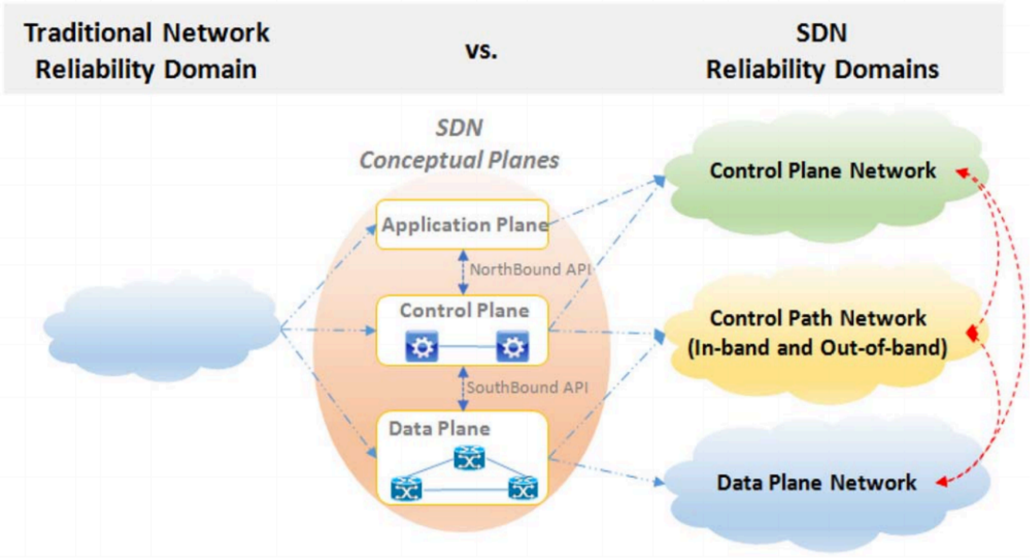


Fig. 1. Multi-lateral SDN reliability domains.

Proposed Solution

Control plane reliability is more crucial.

To address reliability challenges, they propose a control path management framework. Strategies:

- 1) ensure a redundant control connection between the data plane and control plane networks
- 2) virtualize the control plane and control path networks to enable a logically centralized cluster (pool) of controllers.
- 3) a fast and accurate failure detection and isolation mechanism in SDN.
- 4) build a control message orchestration mechanism

Traditional Network Reliablity

TRADITIONAL NETWORK RELIABILITY MECHANISMS

Mechanisms	Protocols
Link bundling	Link Aggregation Control Protocol (LACP) [6], EtherChannel [7]
Multipath routing	Equal-Cost Multi-Path routing (ECMP) [8]
System redundancy	Virtual Router Redundancy Protocol (VRRP) [9], Host Standby Router Protocol (HSRP) [10], Resilient Packet Ring (RPR) [11]
State synchronization	Non-Stop Routing (NSR) [12], Non-Stop Forwarding (NSF) [13], Stateful Switch-Over (SSO) [14]
Failure detection and handling	Ethernet Automatic Protection Switching (EAPS) [15], Ethernet Ring Protection Switching (ERPS) [16], Fast Re-Routing (FRR) [17]

Reliability protocols are embedded in dedicated network devices and treat both data and control failures as interrelated problems according to the physical network topologies.

SDN Reliability

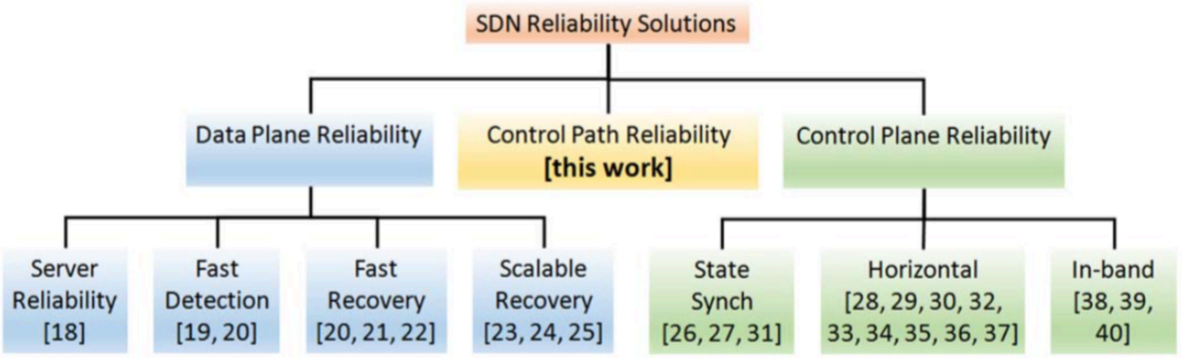


Fig. 2. Classifications of SDN reliability solutions.

[*] uses switch's link signal to check for fast failure detection, faster than controller that identifies failed link through heartbeat messages and sent out an update. No recovery.
[**] Detection and recovery. Extends OpenFlow protocol to support a monitoring function on switches (similar to fault management of MPLS).

- Recover the data plane network from multiple link failures using back-up routes
- Offload control functionality to a switch and achieved fast recovery and better data plane reliability

[*] M. Desai and T. Nandagopal, "Coping with link failures in centralized control plane architecture," in *Proc. IEEE COMSNET*, Bengaluru, India, 2010, pp. 1–10.
[**] J. Kempf *et al.*, "Scalable fault management for OpenFlow," in *Proc. IEEE ICC*, Ottawa, ON, Canada, 2012, pp. 6606–6610.

Control Plane Relialibility

Distributed control plane designs (HyperFlow, ONIX, CPRecovery, B4, and ONOS). Main concerns are scalability and synchronization of network status among multiple physical controllers.

[*] A task manager distributes incoming computations to each controller instance.

These reliability solutions do not consider the correlation between failures of the control plane network and *control path* that are newly introduced in SDN.

Although there are several studies handling the reliability between a controller cluster and a data plane, the path between them has been largely assumed as a **logical connection.**

In this work, *control path* is over a 'network' that can be established as in-band or out-of-band of the existing data plane and controller networks.

[*] Z. Cai, A. L. Cox, and T. S. E. Ng, "Maestro: A system for scalable OpenFlow control," Dept. Comput. Sci., Rice Univ., Houston, TX, USA, Tech. Rep. TR10-11, Dec. 2010.

Availability

Availability is formally defined as the fraction of time that a system is operational (Mean Time To Failure (MTTF)),

To improve availability:

1) increase the uptime of a system (Mean Time Between Failure (MTBF)) or
2) reduce the downtime/outage of a system (Mean Time To Repair (MTTR)).

Little can be done to increase MTTF, since the commodity systems do fail in real operations. Focus reducing MTTR by improving the failure detection and isolation process.

Traditional networks use heartbeat: difficult to identify the exact root cause: absence of heartbeats could have possibly originated from various scenarios of a failure, , thus its recovery mechanism may not be effective.

Traditional Network

Link A fails, *node* 4 detects failure on *port* 1, also means a failure on *node* 2's *port* 2, *node* 2 itself, or a link in-between.

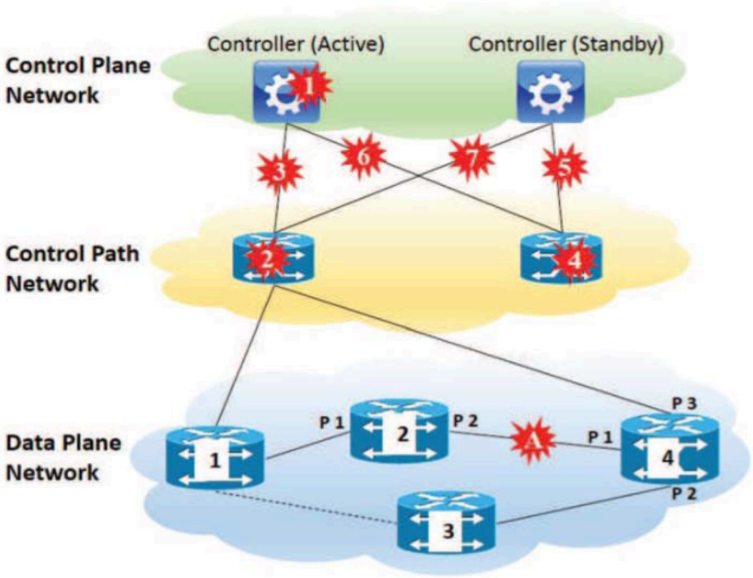
node 2 also detects failure on *port* 2, also means a failure on *node* 4's *port* 1, *node* 4, or link. Each node blocks the failed ports.

In L2 network, a spanning tree will be regenerated. Network will be blocked until the convergence.

In L3, the failure information will be propagated to other nodes via routing protocols. The L3 routes will be recalculated.

SDN

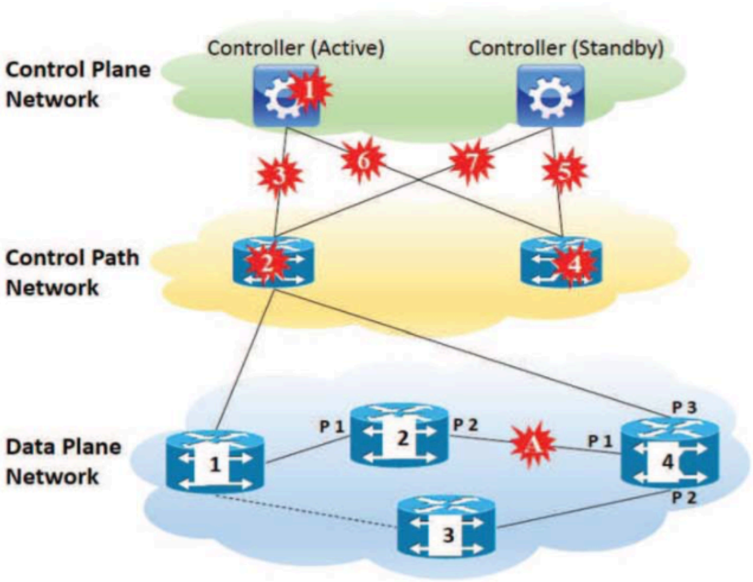
Node 1 and 2 will not send echo. controller sees potential node failures on both *node* 1 and *node* 2 and sends an LLDP (Link Layer Discovery Protocol) message to regenerate the network topology. According to a new network topology, the SDN controller reassigns the data flow tables. Meanwhile, all the data packets and control messages forwarded to the failed link will be dropped (or cached depending upon the implementation).



(a)

Illustration of heartbeat-based SDN reliability models: (a) SPoF on *control path* (b) no SPoF on *control path*.

UCDAVIS



(a)

SDN

Node 1 and 2 will not send echo. Controller sees potential node failures on both *node* 1 and *node* 2 and sends an LLDP (Link Layer Discovery Protocol) message to regenerate the network topology.

According to a new network topology, controller reassigns the data flow tables. All data packets and control messages forwarded to failed link will be dropped (or cached depending on the implementation).

Traditional Network

Link A fails, *node* 4 detects failure on *port* 1, also means a failure on *node* 2's *port* 2, *node* 2 itself, or a link in-between.

node 2 also detects failure on *port* 2, also means a failure on *node* 4's *port* 1, *node* 4, or link. Each node blocks the failed ports.

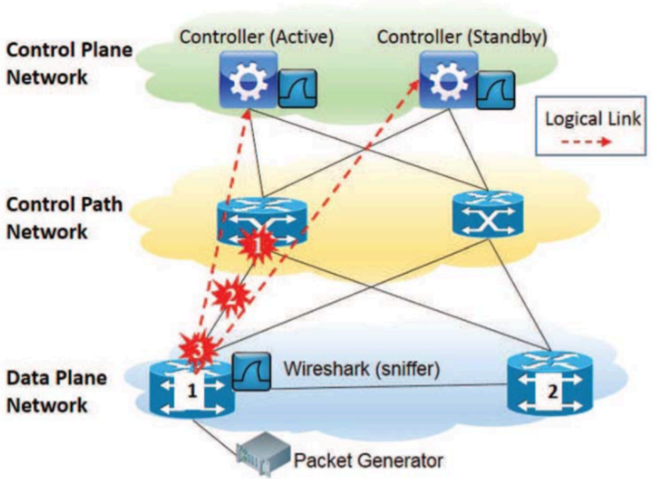
In L2 network, a spanning tree will be regenerated. Network will be blocked until the convergence.

In L3, the failure information will be propagated to other nodes via routing protocols. The L3 routes will be recalculated.

Illustration of heartbeat-based SDN reliability models: (a) SPoF on *control path* (b) no SPoF on *control path*.

SDN Reliability Challanges: Observations

A. SPoF With Multiple Logical Control Path Connections



Recovers from controller failure

Fig. 4. Illustration of unnecessary single points of failure (see the yellow numbers): Multiple logical connections are overlapped such as (1) the legacy switch between the controller and the OpenFlow switch, (2) the link between the legacy switch and the OpenFlow switch, and (3) the interface of the OpenFlow switch.

- network may experience not only a long recovery time but ultimately a service disruption as well.
- effectively **disperse** logical connections to fully exploit available physical redundancy, so that a failure detection and a switch-over would take place seamlessly without requiring a re-connection process.

UCDAVIS

B. Configuration of Multiple Controllers

have multiple concurrent logical connections from switches to multiple controllers to minimize switch-over time.

Current OpenFlow switch has to know its controllers. No dynamicity. Manual configuration per switch for the changes in control cluster.

Management Cost = $$\sum_{Time} \sum_{c}^{\#ofcontrollers} (1 - P_c)$$

where *Pc* denotes the probability controller topology can last without any changes (add/remove controller)

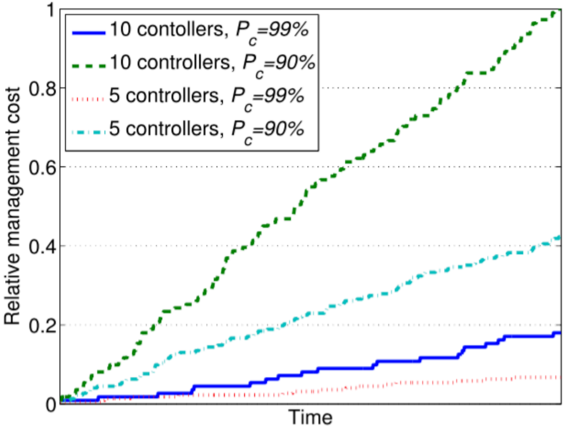


Fig. 6. Relative management costs for given network sizes: the management cost increases as the number of OpenFlow switches and the probability of a cluster configuration change increases.

C. Unrecoverable Control Path Failure Case

Slave only receives port-status messages but not packet-in/flow-removed.

Slave is able to detect the network topology/status changes.

if a slave does not receive heartbeat messages consecutively, it initiates a process to become master controller, sends a role change request message to switches.

However, if slave keeps receiving heartbeat messages from master while the *control path* towards the OpenFlow switches is in a failure mode.

C. Unrecoverable Control Path Failure Case (cont.)

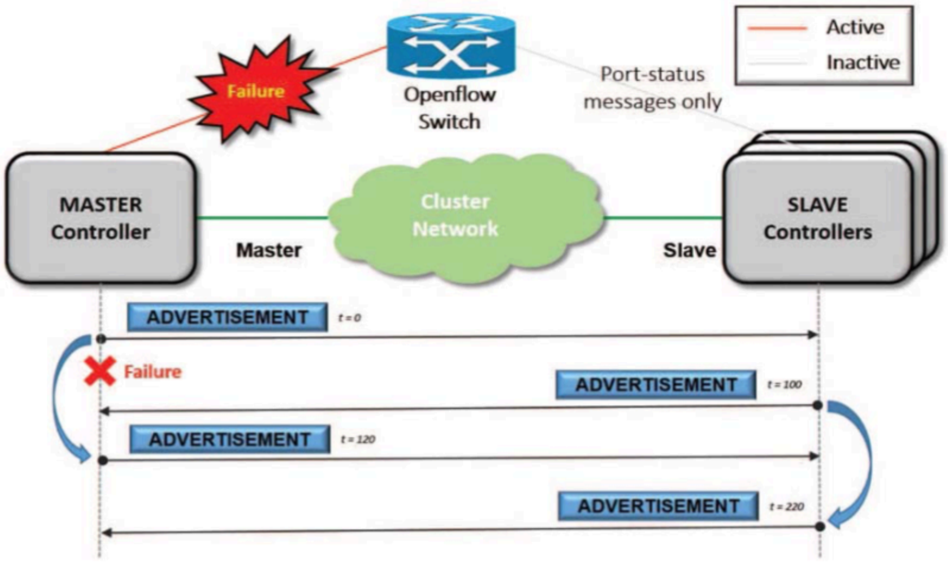


Fig. 7. Scenario when an OpenFlow switch loses its master controller: the connection between the slave controller and the OpenFlow switch transfers only port-status messages.

Master and slave controllers are connected through a legacy switch for synchronization

Current specification does not allow an OpenFlow switch to initiate its controller's role change.

This is because current reliability feature does not consider the correlation between failures of the *control plane network* and *inter-connection network*.

UCDAVIS

D. Control Message Scalability Issues

SDN imposes excessive control traffic overhead, controller platforms allow a variety of heterogeneous application interfaces and protocols to the data plane.

It can cause various scalability and reliability problems: slow message processing, potential message drops, delayed root cause analysis, and late responses against urgent problems.

In traditional routers, internal packet prioritization is used.

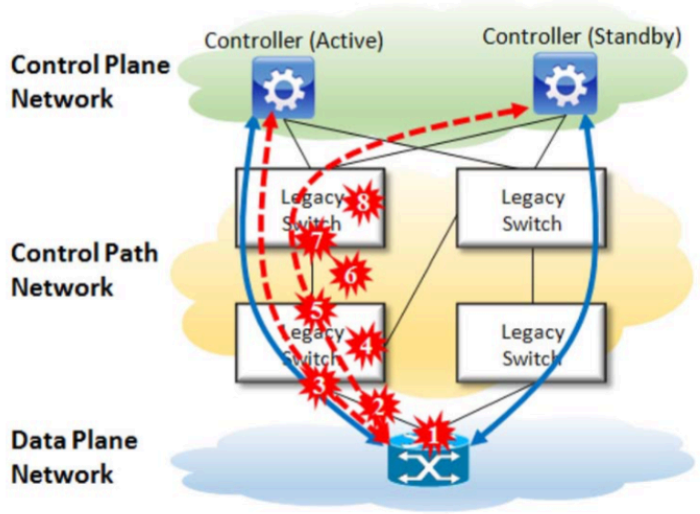
Current OpenFlow specification, the SDN controllers drop packets randomly regardless of the importance and urgency of the packets.

PROPOSED SDN RELIABILITY MANAGEMENT FRAMEWORK

A. Aligning Logical and Physical Control Path Redundancy

Route logical connections through physical disjoint paths to alleviate/remove the SPoF problem.

Modified the OpenFlow reference implementation by adding an interface selection feature.



UCDAVIS

B. Controller Cluster Structure Agnostic Virtualization

Each switch does not have to know the distinct IP addresses or port numbers of controllers.

To automate adaptation to control plane changes:

- Virtualize physically distributed multiple controllers into one logically centralized controller with 1 virtual IP address.
- Associate Virtual IP with **cluster information broadcaster** (***CIBroadcaster***). *CIBroadcaster* will send the up-to-date cluster information to new switches.
- Other controllers are backup broadcasters and listen to the heartbeat messages from CIB.

B. Controller Cluster Structure Agnostic Virtualization (cont.)

How a controller cluster maintains consistency of the cluster information: Hello and update messages.

C. Fast and Accurate Failure Detection and Recovery

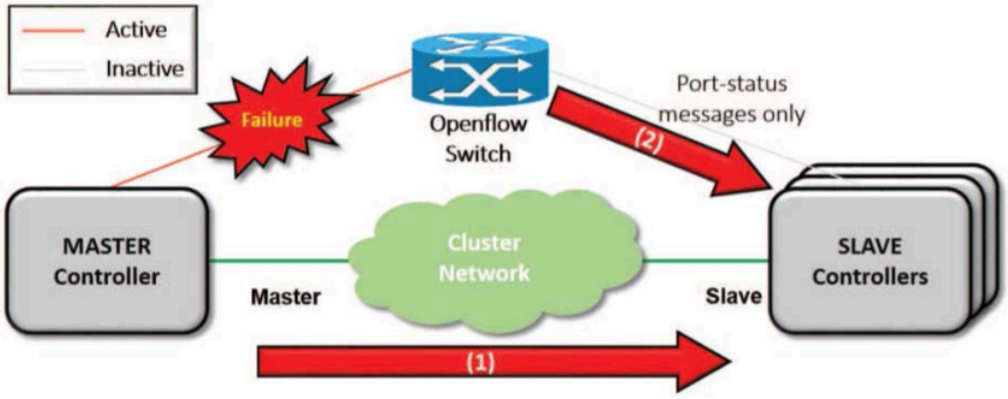


Fig. 12. Fast and accurate failure detection and recovery using topology awareness and link signals: (1) the master controller initiates the recovery (Algorithm 3) (2) the OpenFlow switch initiates the recovery (Algorithm 4).

C. Fast and Accurate Failure Detection and Recovery (cont.)

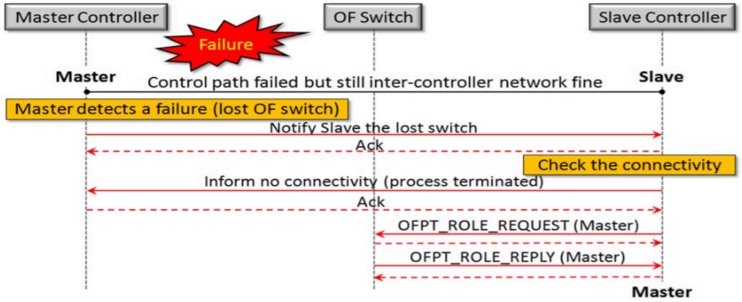


Fig. 13. Failure recovery initiated by the master controller (Algorithm 3).

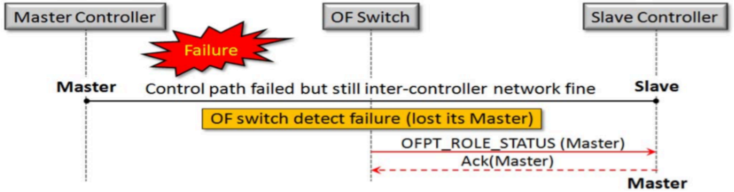


Fig. 14. Failure recovery initiated by an OpenFlow switch (Algorithm 4).

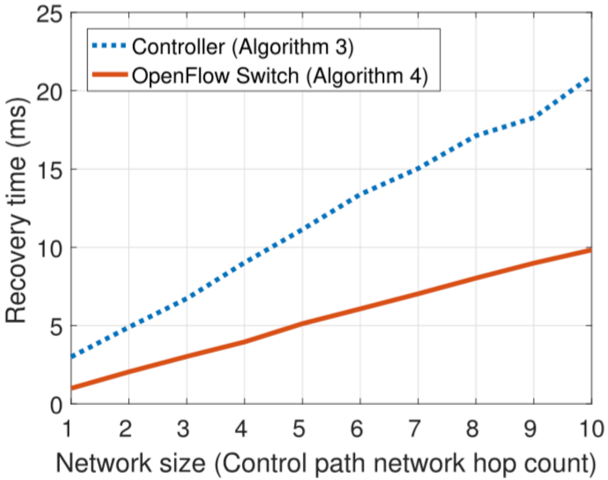
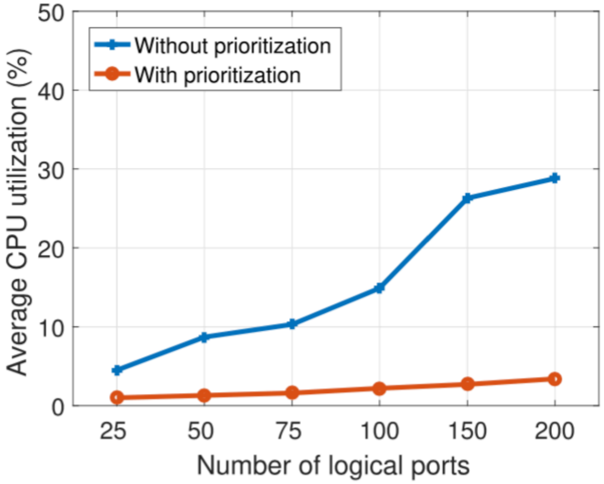


Fig. 15. Comparison of recovery schemes initiated by an OpenFlow switch and a controller.

UCDAVIS

D. Control Message Orchestration Module

classification/prioritization system for creating, handling network control messages.



Impact of prioritization on CPU utilization.

Set 2 bits of the type of service (ToS) field in IPv4 header according to importance of classified control message.

Hence, controllers and switches can differentiate processing sequence and selectively drop received control messages.