# DISASTER-RESILIENT CONTROL PLANE DESIGN

Sedef Savas

December 16th, 2016

Netlab Friday Group Meeting

# Control Plane Mapping Problem

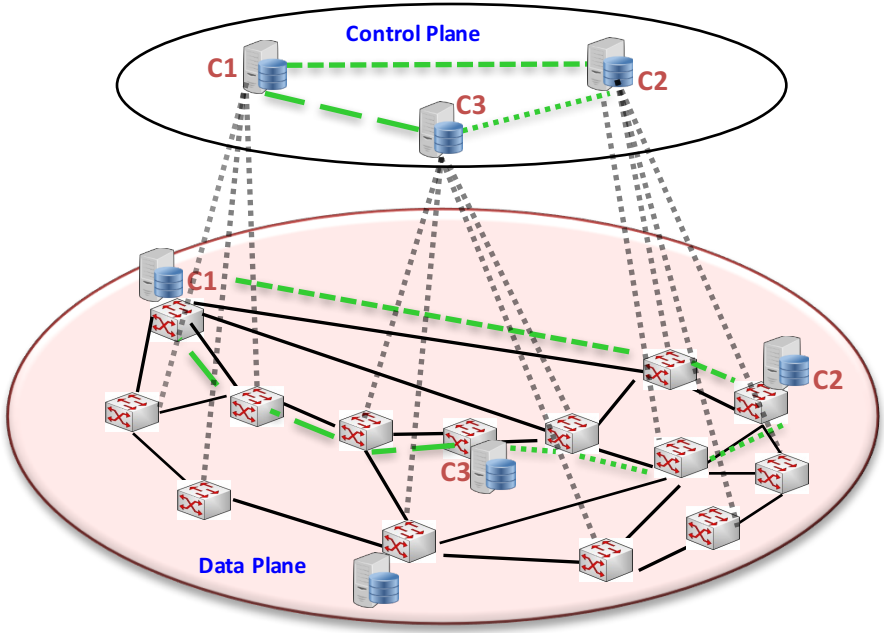
Design a resilient control plane that satisfy latency constraints.

A distributed control plane can be designed as an overlay (i.e., virtual/logical) network mapped over a physical (i.e., backbone) network.

- virtual nodes where controllers are located and virtual links connects them.

**We propose a survivable control plane mapping scheme to ensure control-plane connectivity against both single point of failures and large-scale disaster failures in SDN.**

# Control Network Design and Mapping




- Determining # of controller and their placements ✓
- Determining logical control plane topology ✓
- Mapping control plane to physical network ✓
- Controller-switch assignment ✓

# Recap – Related Work

- Protecting switch-controller communication.
- Not provisioning/protecting controller-controller communication.
- Controller placement and switch controller assignment techniques.
- No disaster/multi-element failure scenario.
- There exist only a finite number of distinct regional failures in a given geographical area. Assume a set of possible regional failures, which is given in advance, at most one of these regional failures that is active at any time.

# Recap (cont.)

- Assign multiple paths to single controller, assign multiple paths to multiple controllers.
- Minimize # of controllers necessary to provide latency guarantees.
- Load balancing.
- Less hops (switch to controller), more reliable.
- Maximizes number of node disjoint paths between controllers and assigned switches.
- How many controllers (minimize) the node should connect to in order to achieve ”five nines reliability”?

# Why designing a VN for control plane?

Reprovisioning is always an option for data plane as long as control plane is up.

The time of need, there may not be enough resources, tearing down and setting up takes time. Data plane cannot work properly.

As long as, we have a connected control plane, system will be up and running.

All needs to be connected with all.

# Why design as a VN?

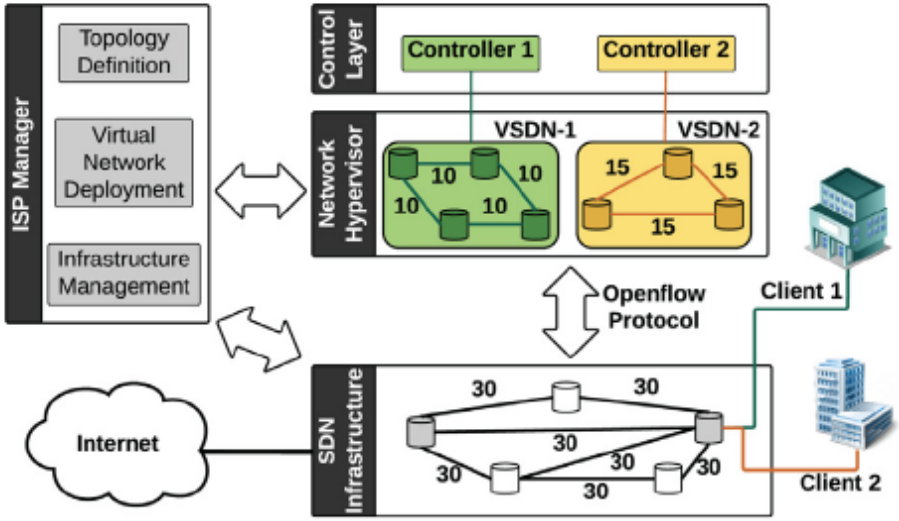
Paper proposes Bw-Risk-Ratio algorithm to deploy VSDNs with best ratio between network reliability and bandwidth allocated to VSDN.

Allocate the shortest path with higher available bandwidth and lower failure risk.

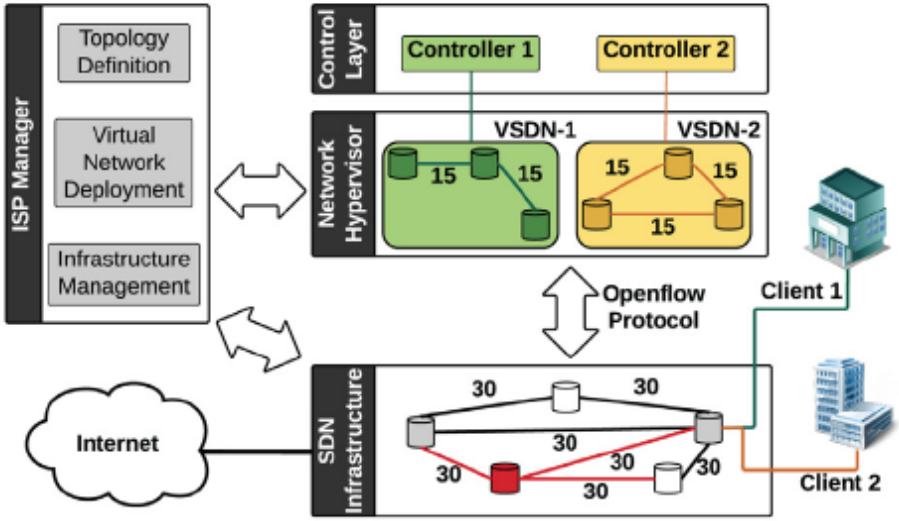
**Network resilience:** capacity of network to provide a minimum specified level of service in situations of faults in standard operation.

VSDN allocation according to reliability, aims to minimize total bandwidth committed to solve requests.

The network hypervisor enables a virtual network to be independently managed by a controller and to be dynamically provisioned



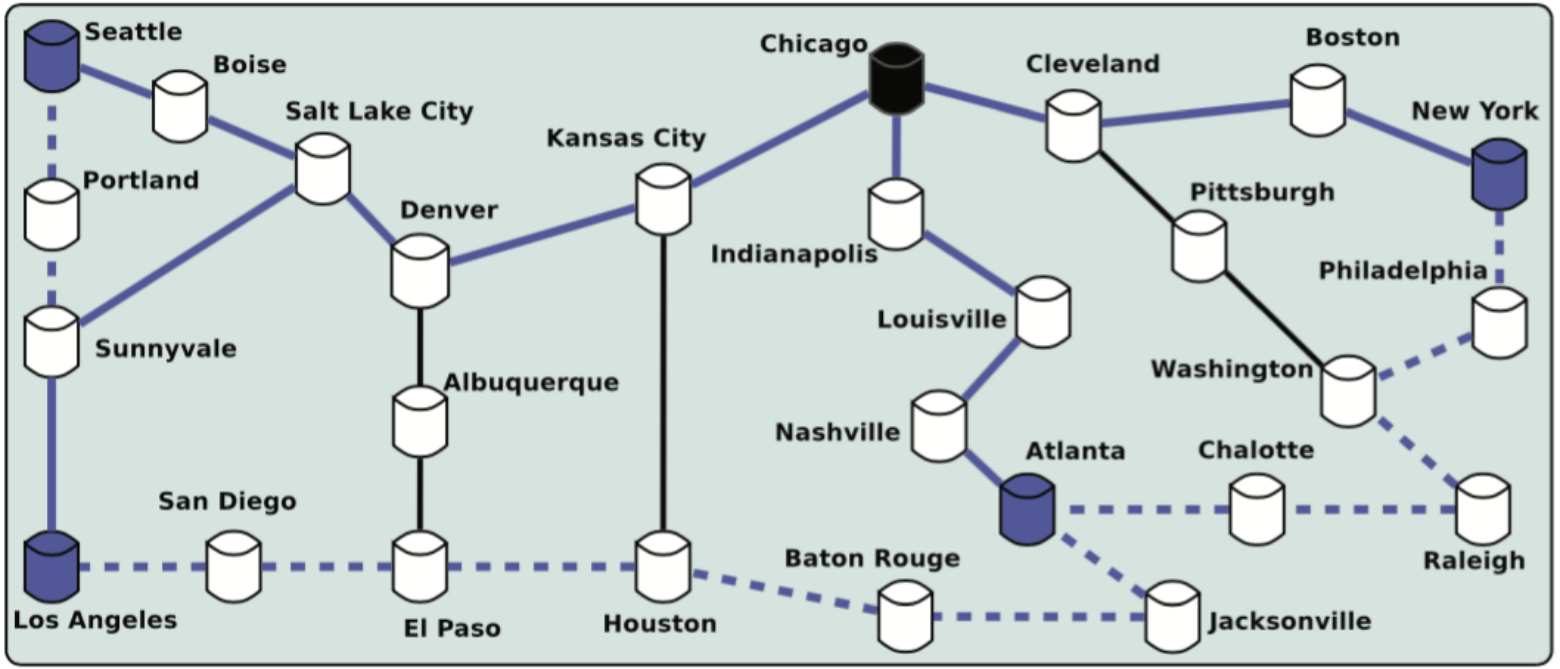

(a) Normal operation.




(b) Operation under failure event.

**black node:** the client/root node and nodes in blue illustrate the destination nodes.
**blue lines:** the links allocated in scenario of no redundancy.
**blue dashed lines:** additional links allocated in full redundancy situation.

**VN is defined as set of requests**




Rafael L. Gomes, Luiz F. Bittencourt , Edmundo R.M. Madeira, Eduardo Cerqueira, and Mario Gerla, Bandwidth-aware allocation of resilient Virtual Software Defined Networks, Communication Networks, 2016.

UCDAVIS

# Risk modeling

$$Risk_{Score} = \frac{\sum_{i=1}^{n} RiskFactor_i}{n}.$$

Rules.

| Occurrence | Operation | Impact | Risk factor |
|---|---|---|---|
| High | And | High | High |
| High | And | Medium | High |
| High | And | Low | Medium |
| Medium | And | High | High |
| Medium | And | Medium | Medium |
| Medium | And | Low | Low |
| Low | And | High | High |
| Low | And | Medium | Low |
| Low | And | Low | Medium |
| None | Or | None | None |

Risk model information.

| Region | Nodes | Risk events | Risk score |
|---|---|---|---|
| A | Seattle, Portland, Sunnyvale, Los Angeles, and San Diego. | Meteorological and geophysical. | 1.5 |
| B | Boise and Salt Lake City. | Meteorological and climatological. | 1 |
| C | Albuquerque and El Paso. | Meteorological. | 0.75 |
| D | Kansas and Denver. | Meteorological and climatological. | 1 |
| E | Houston, Baton Rouge, and Jacksonville. | Meteorological and hydrological. | 1.5 |
| F | Chicago, Indianapolis, and Louisville. | Meteorological, hydrological, and geophysical. | 2.25 |
| G | Nashville, Atlanta, Charlotte, and Raleigh. | Meteorological and hydrological. | 1.5 |
| H | Cleveland, Washington, Boston, New York, and Philadelphia. | Meteorological and hydrological. | 1.5 |

# Traditional protection vs Survivable VN

Allocating requests as VN better perform than K shortest disjoint path method to meet reliability requirements.

Even in a star topology, less bw is used, higher successful VN allocations, and higher connectivity status under failure events.

Gain is much less than a case where all nodes need to communicate with all others.

# Difference from classical VNE approaches

- Virtual network topology is not given. (nodes and links unknown)
- # of nodes is not defined.
- Extra requirements:
    - ensure switches have at least 1 VN node in reachability distance in any case.
    - Different latency definition. Node-node latency, we do not only considers VN's latency.

Existing work: initial VN request mapping, backup substrate nodes, backup substrate paths, VN request migration.

If any node is in a given regional failure, add another node.

# Survivable VN Mapping

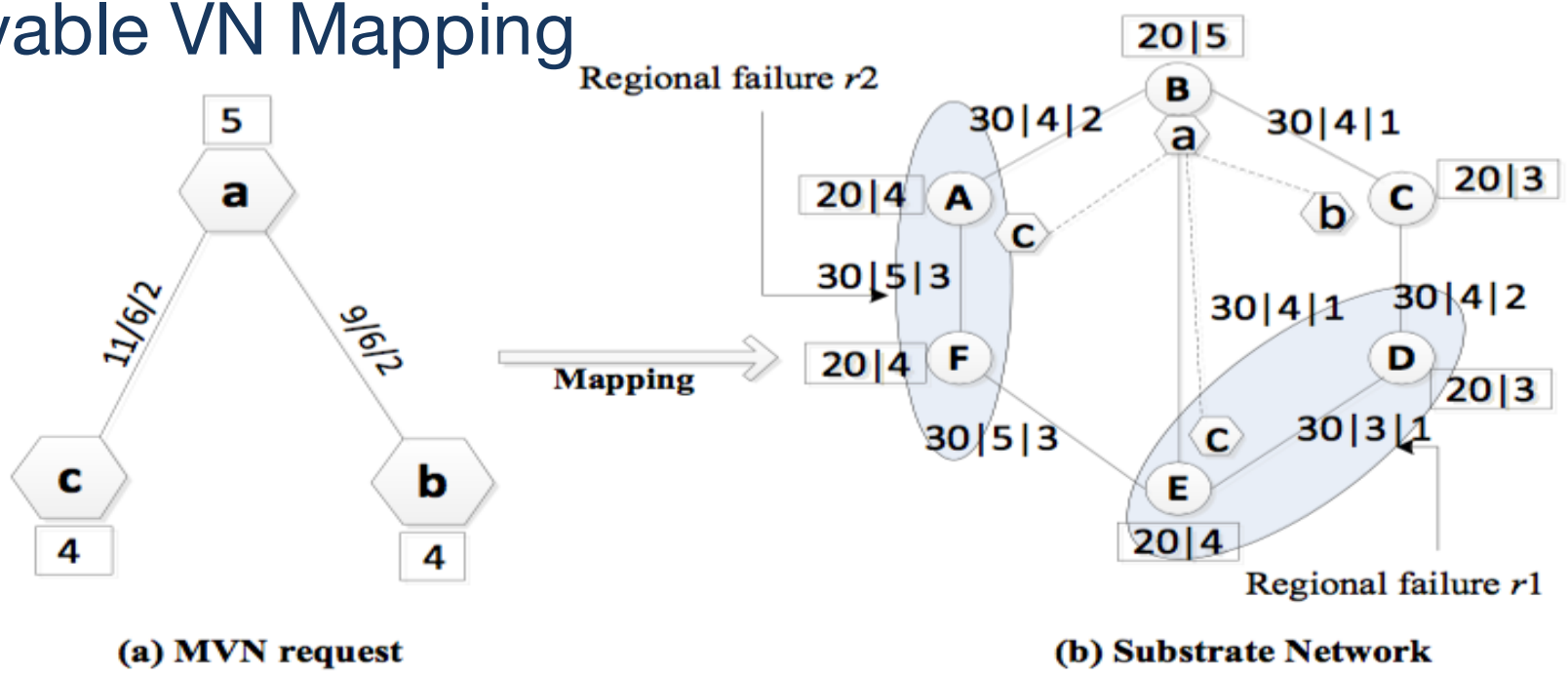



**Fig. 2** Example of MVN mapping under regional failure(s)

Only one regional failure occurs at a time, the resource reserved on substrate nodes and substrate links can be shared among different failure scenarios



D. Liao, G. Sun, V. Anand, and K. Xiao, **"Survivable Mapping for Multicast Virtual Network under Single Regional Failure", Globecom, 2014.**

# Limitations

Failover mechanisms require information in advance, which, in turn, has been overlooked.

Do not decide on number of controllers.

No specific controller locations.

No latency limit while selecting controllers and paths. Primary path length do not differ much, but what about backup path? (Disjoint)

# Objective for reliable control plane design

**High availability** and **low control plane latency** are necessary to guarantee data plane performance.

Minimize # of controllers.

Minimization of the worst case latency between switches and controllers.

Minimization of inter controller latency.

# Why minimize controllers?

- Distributed controllers exchange information about network state in a timely manner and maintain accurate global overview (loads as well).

- Many VSDNs will coexist in an SDN-enabled physical infrastructure. Hence, there will be numerous distributed control planes.

- **Synchronization cost:** Shortest path between the farmost controller pair.
- Synchronization methodology: Flooding. Aggregate incoming data and pass it to neighbors.
- Also maintenance is harder. Higher risk of loosing a controller.
- More capacity usage, not necessarily more bandwidth usage.
- Less controller, higher chance that the controller do not need to forward rule installing requests.
- Minimize bandwidth usage in a multi-tenant network with VSDNS. Includes synch. cost + flow setup cost.

# Problem formulation

**Given:** Topology, Datacenter locations, Disaster size

**Objective**

Find minimum # of controllers, place them, connect them, assign switches to them.

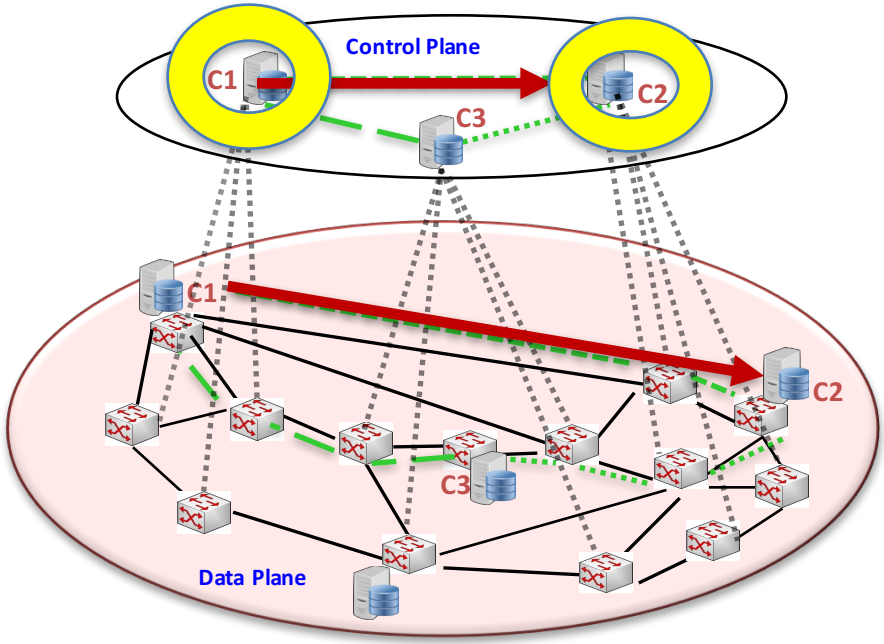
# Constraints

**Latency requirements:** Limits worst case.

1. Maximum latency between switches and controllers.
2. Maximum latency between any controller pair, affects synchronization time.
3. Maximum path setup latency.

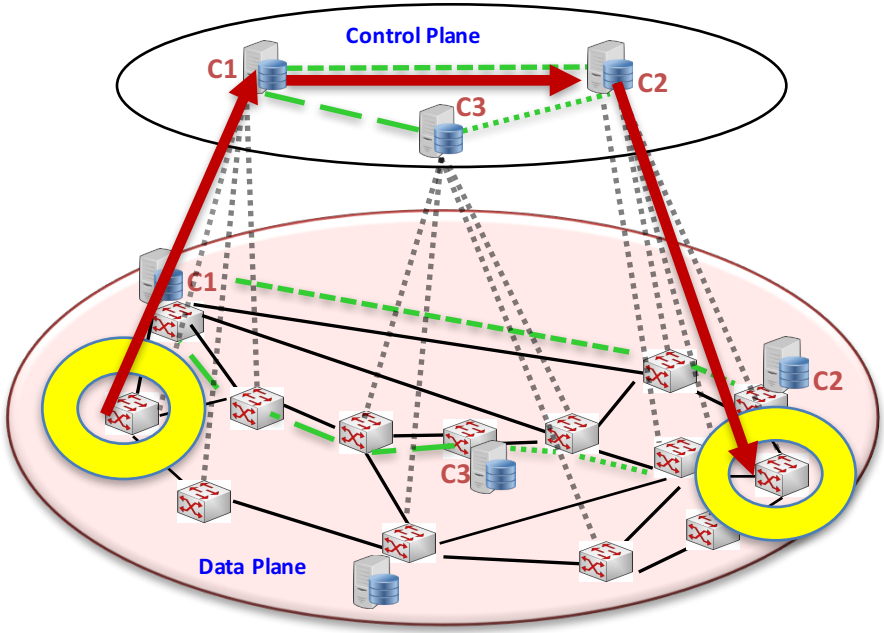
Switch controller latency's affect: For all possible disasters, after failure, make sure there is a path to a controller within latency limits. This will be preprocessed.

# Synchronization latency




Periodically, flooding state updates.
Not every controller sends every other controller the same info.

# Worst case path setup latency



# Controllers: close to switches or other controllers?

**Switch-controller communication:**

- Periodic state update
- New flow setup

**Controller-controller communication:**

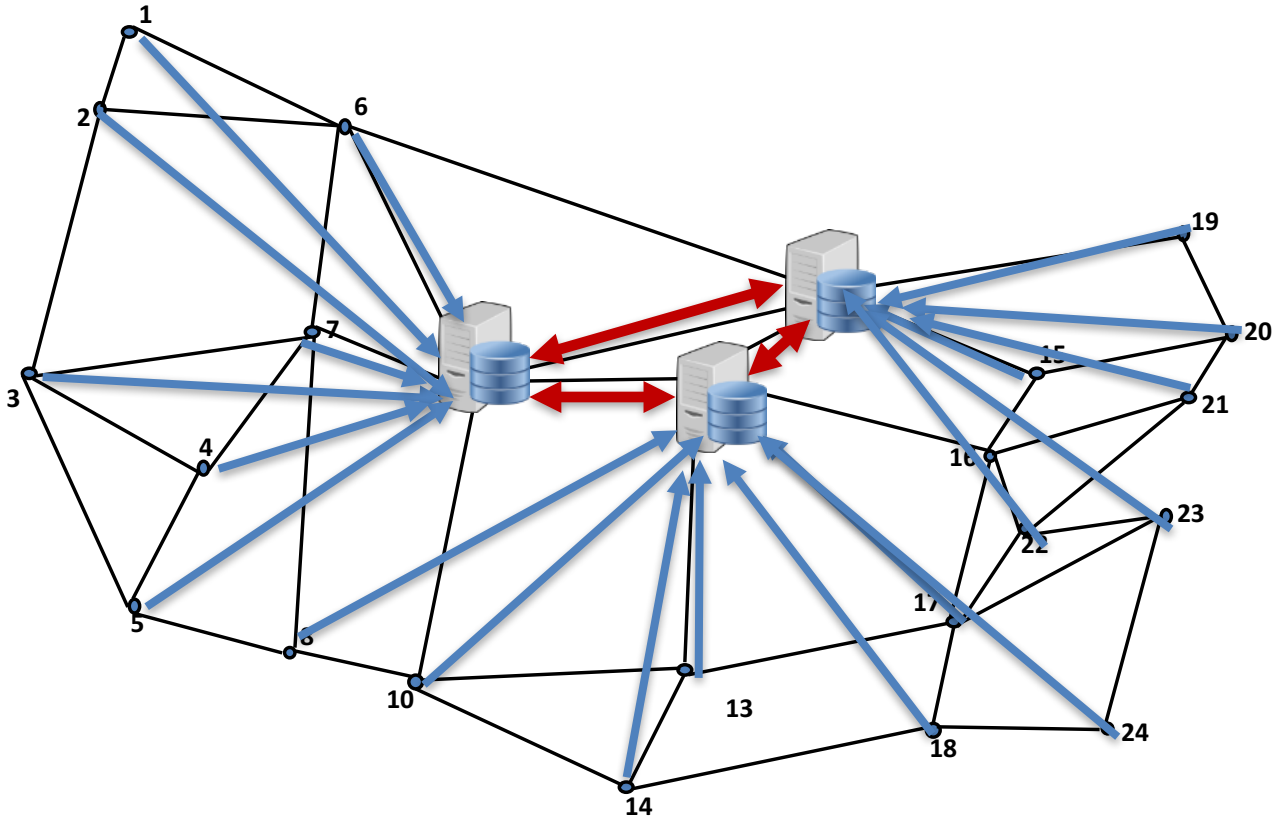
- Synchronization
- New rule installation requests

**Depends on the topology and constraints:**

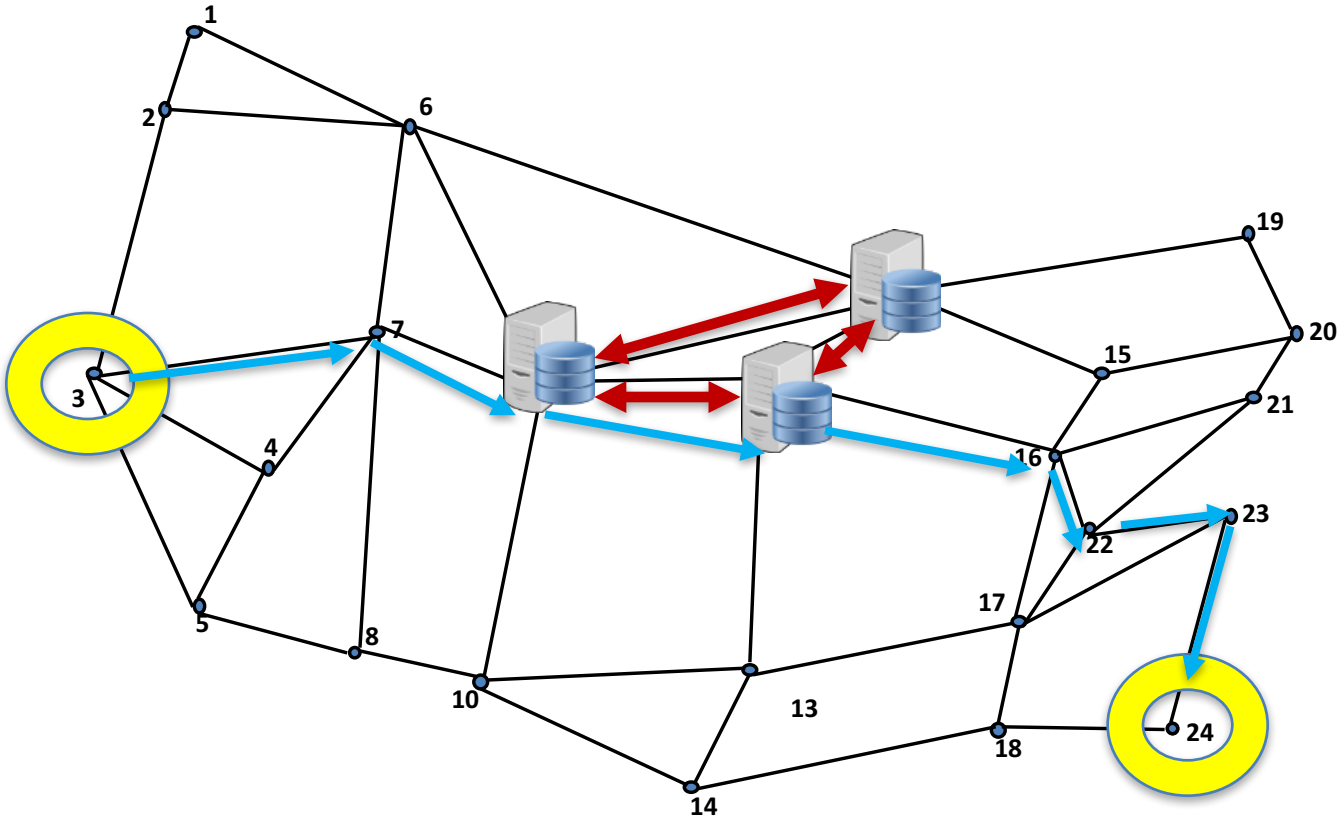
Many switches connect to a single controller: Controllers should be closer to switches in many switches with high loads scenario. More flows do not need to be sent to other controllers, but routed within the cluster. Decrease flow setup latency.

If not much can be gained from placing controllers apart, then place them close. Decrease synchronization.

# Close to other controllers

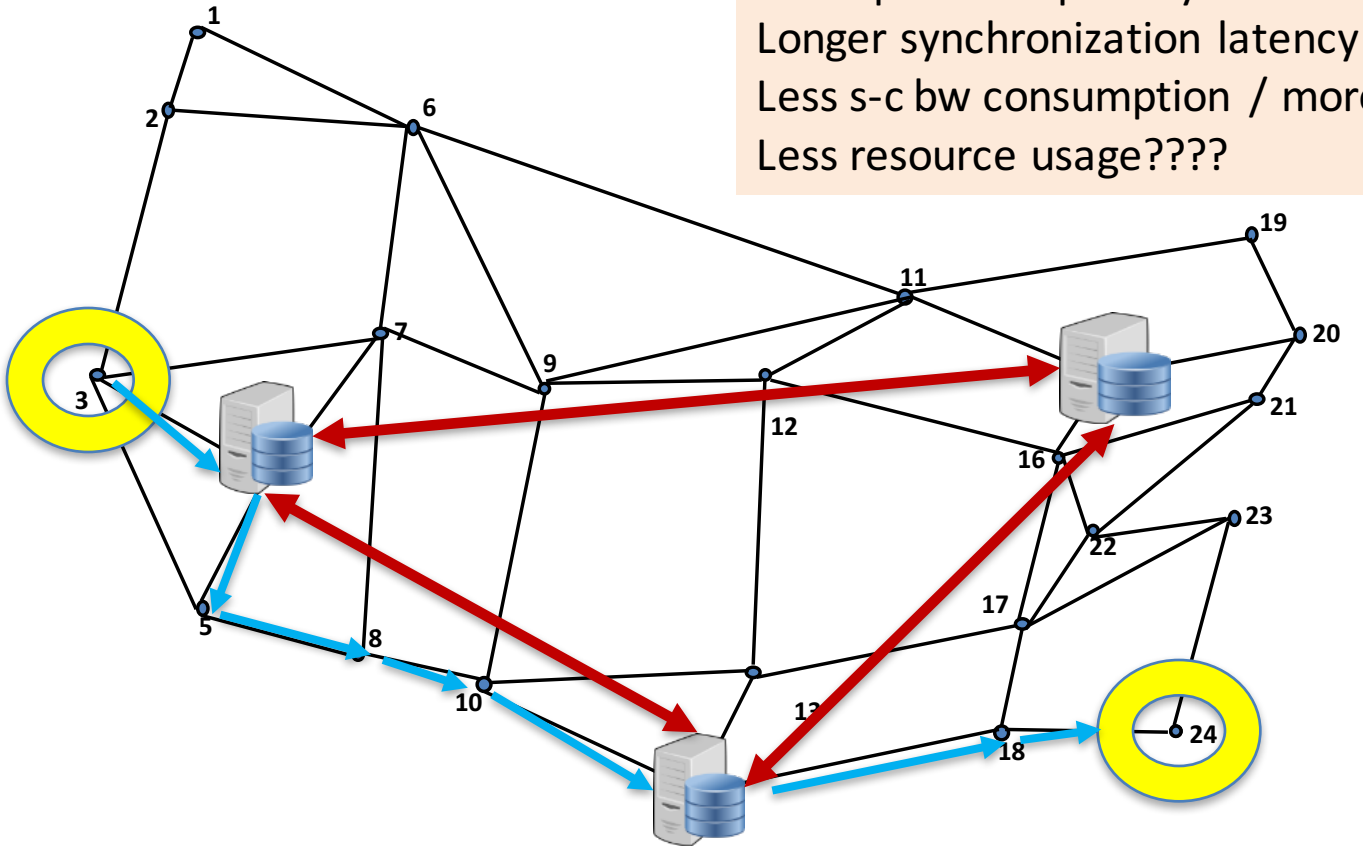


# Close to other controllers

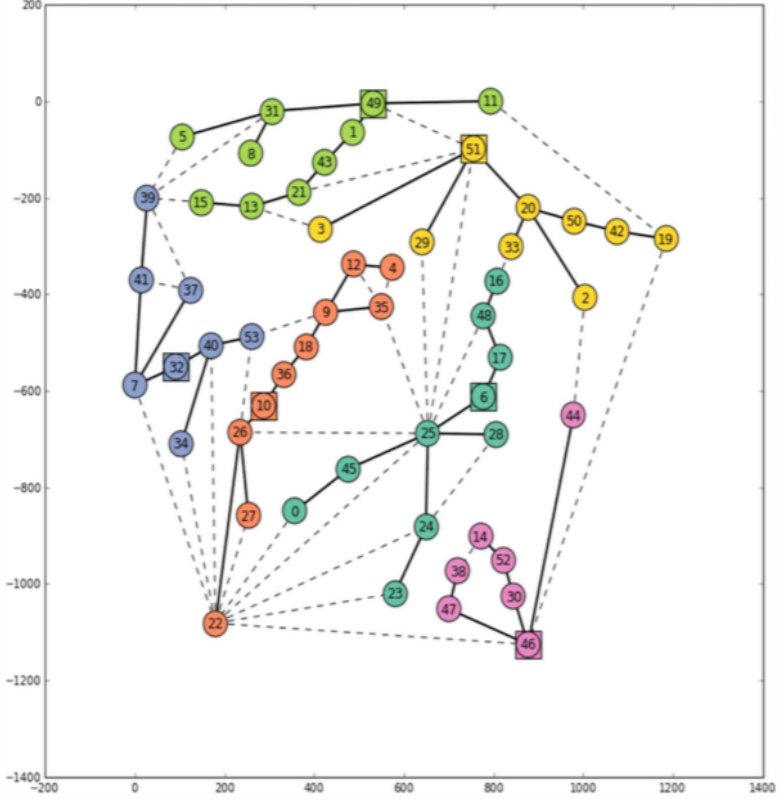


# Close to switches

Same path setup delay
Longer synchronization latency
Less s-c bw consumption / more c-c (negligible)
Less resource usage????




Max latency between router-controller(30% of the graph diameter) and controller-controller (70% of the graph diameter) is set.



ig. 6. Optimal solution with $l_{max} = 30\%$, $l_{cc\text{-}max} = 70\%$ and $\delta = 3$

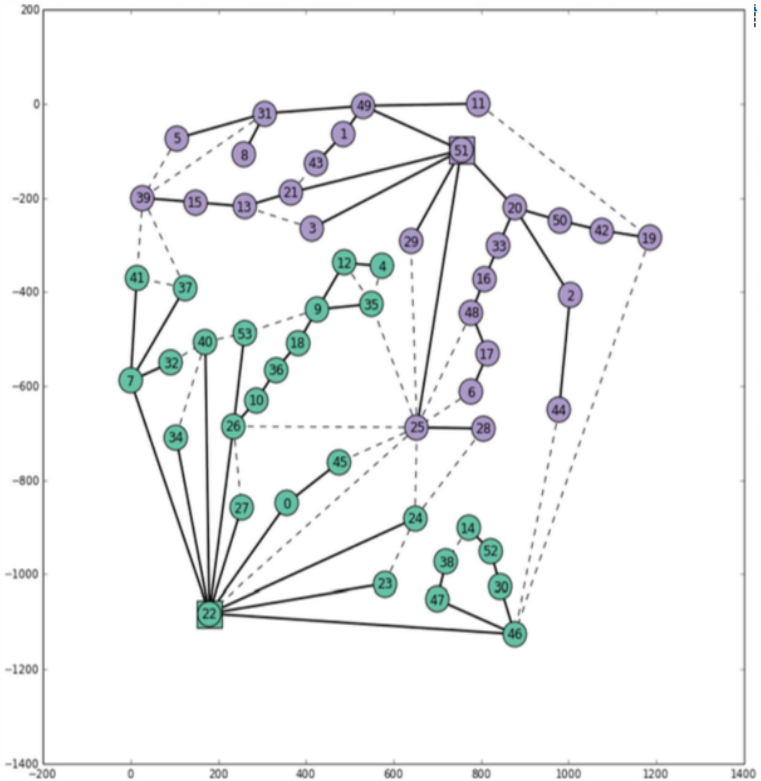


Fig. 7. Optimal solution with $l_{max} = 50\%$, $l_{cc\text{-}max} = 70\%$ and $\delta = 3$

TABLE I. SENSITIVITY ANALYSIS OF THE CONSTRAINTS ON THE OPTIMAL SOLUTION

| $l_{max}$ | $l_{cc\text{-}max}$ | $\delta$ | Placement |
|---|---|---|---|
| 0.4 | 0.7 | 3 | 4 21 |
| 0.3 | 0.7 | 3 | 9 18 21 28 |
| 0.27 | 0.7 | 3 | 2 4 11 26 28 34 |
| 0.25 | 0.7 | 3 | 2 9 18 22 23 24 26 30 |
| 0.35 | 0.4 | 3 | $\emptyset$ |
| 0.35 | 0.7 | 3 | 10 19 22 29 |
| 0.35 | 0.9 | 3 | 14 16 19 32 |
| 0.35 | 1 | 3 | 2 5 8 9 |
| 0.35 | 0.7 | 1 | 2 3 5 9 |
| 0.35 | 0.7 | 7 | 6 9 28 |
| 0.35 | 0.7 | 15 | 6 31 35 |

# Constraints (cont.)

**Capacity requirements**

Datacenters have capacity limit. Each controller will be responsible of a limited # of nodes.

**Resiliency/Connectivity requirement:**

- Control plane resilient against single point of failures. At least 2-connected. Depending on the disaster range and the topology more may be needed.
- After any disaster at size r, alive controllers stay connected. And all switches can be assigned to a switch within latency constraint.
- Initially, at least 2 controllers within latency requirement of switches to achieve these.

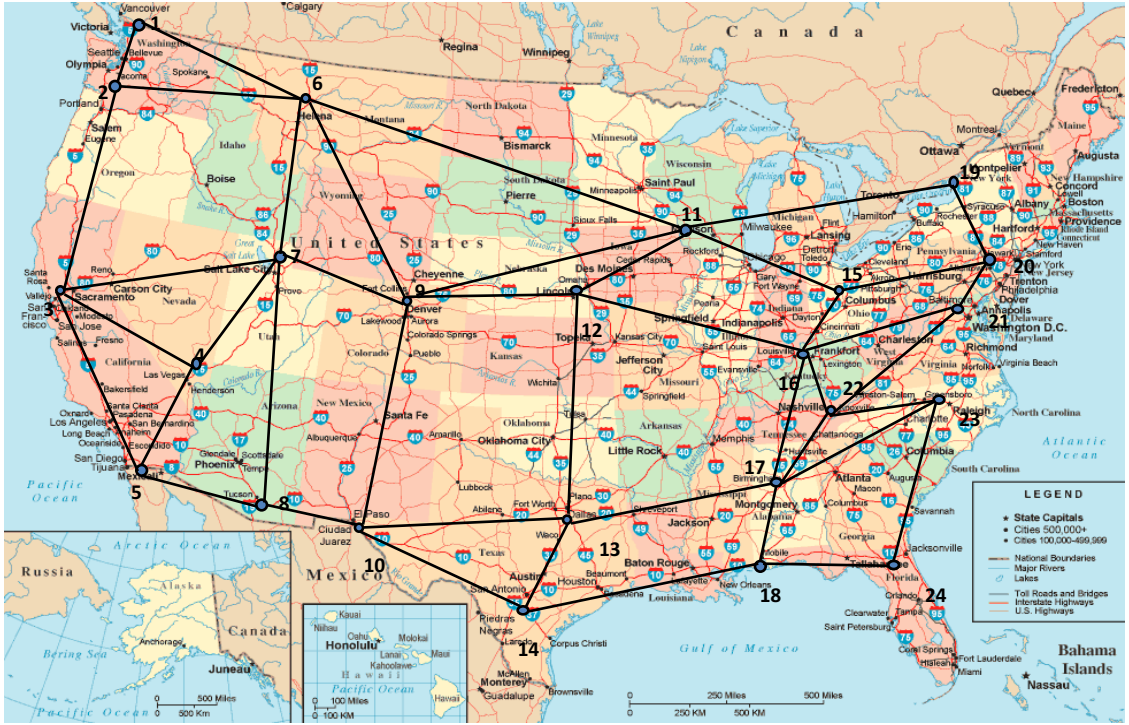
# Assumptions

- Uniform demand.
- Only specific nodes can be controller locations.
- Each router to exactly one controller.
- All switches being controlled by their nearest controller and all control paths being the shortest paths between the switch and the assigned controller.

- Do not consider backups between switches and controllers. As long as control plane is up and physical layer is connected, control plane will reach unattached switches. **Hard to find disjoint paths that will survive all r-sized disasters.**

Do not consider reassignments in case of disasters, one disaster at a time, no need to consider normal mode of op. constraints (only latency), so not much to show. **Design problem.**

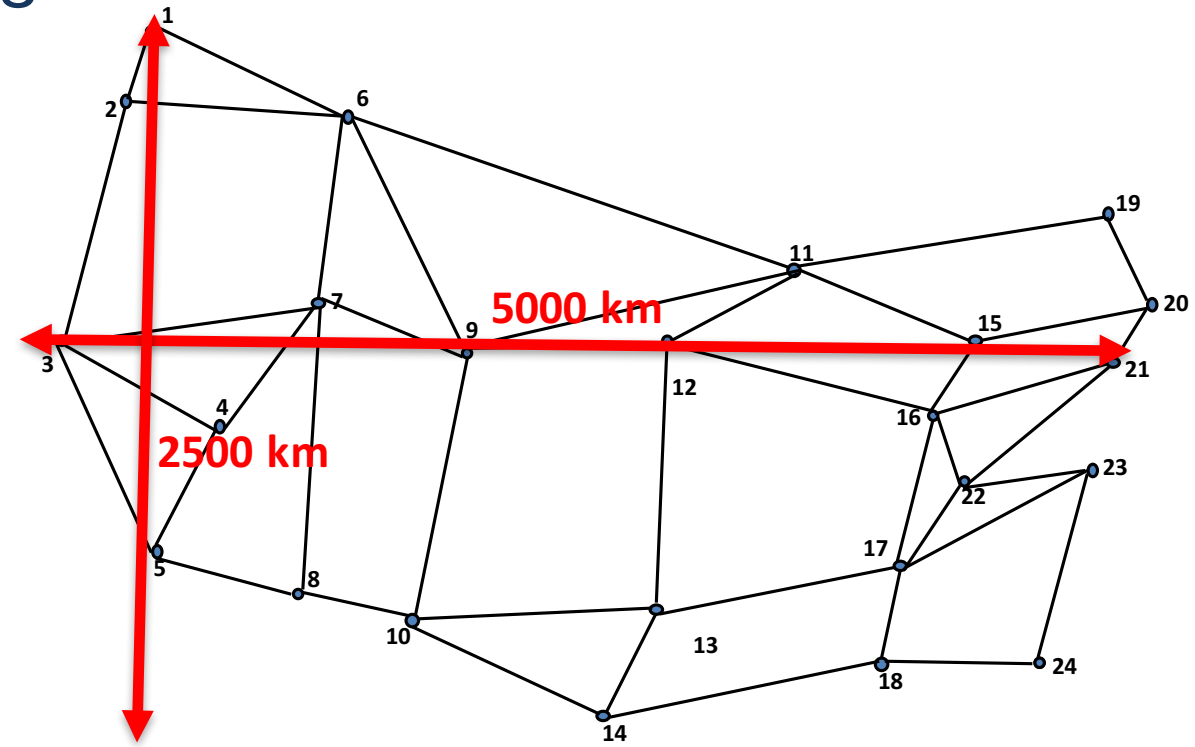
# Modeling disasters



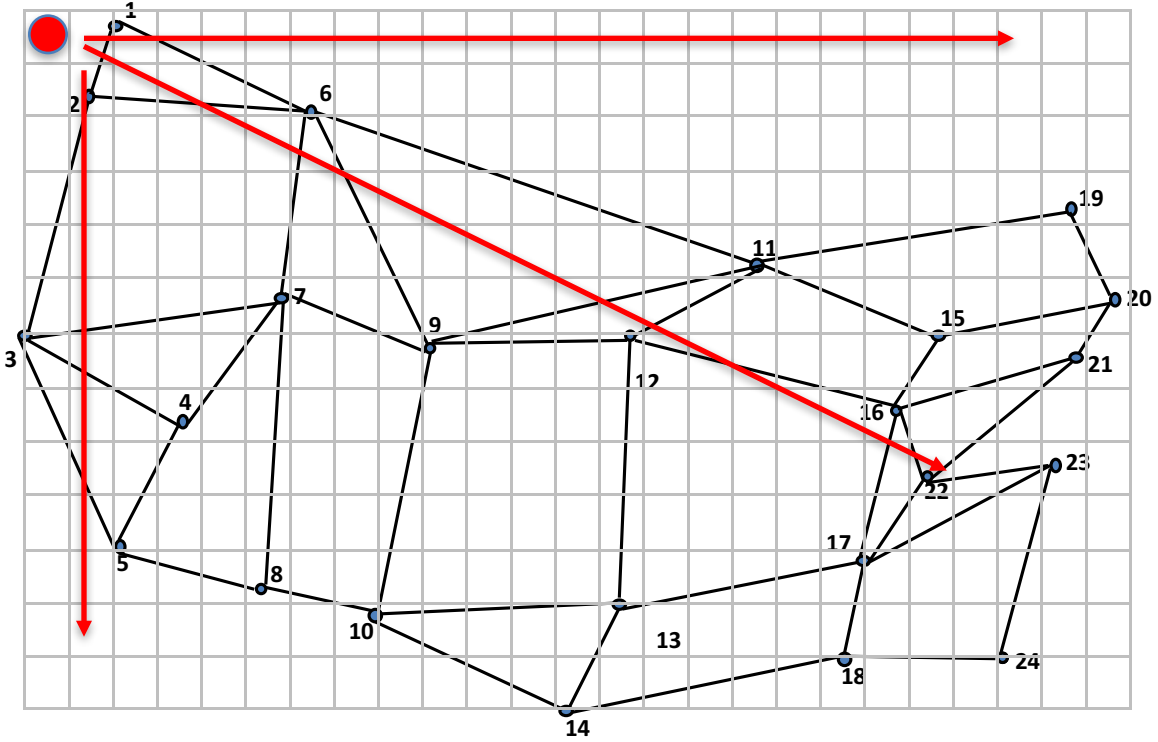



UCDAVIS

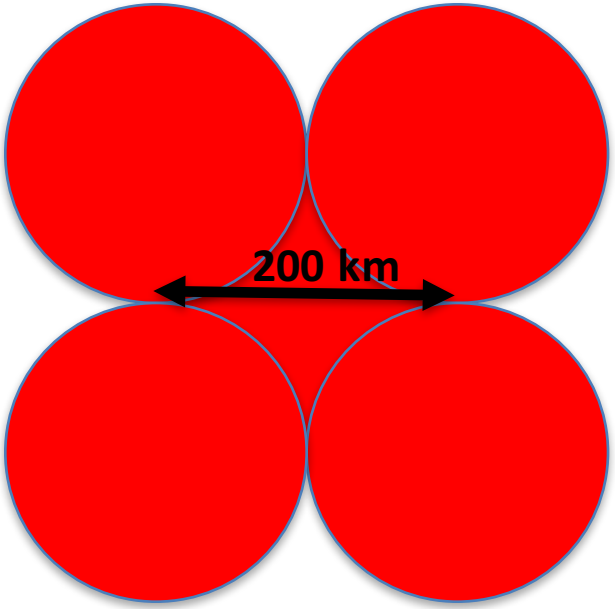
# Modeling disasters

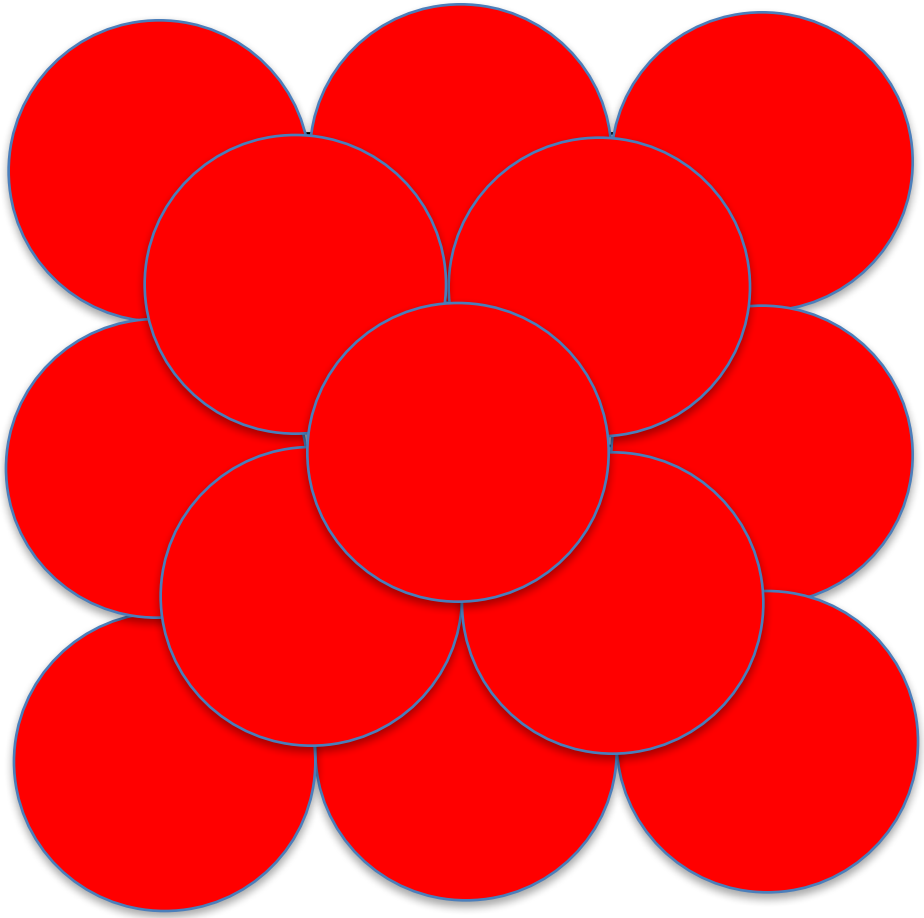


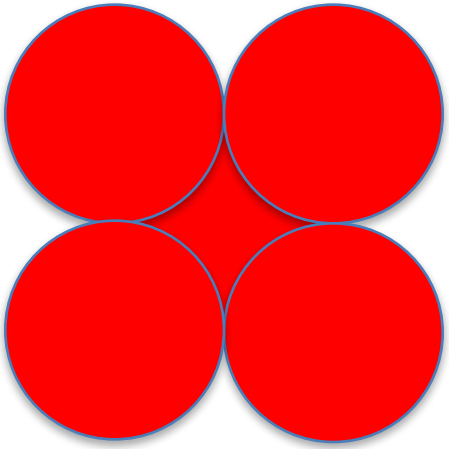
# Modeling disasters



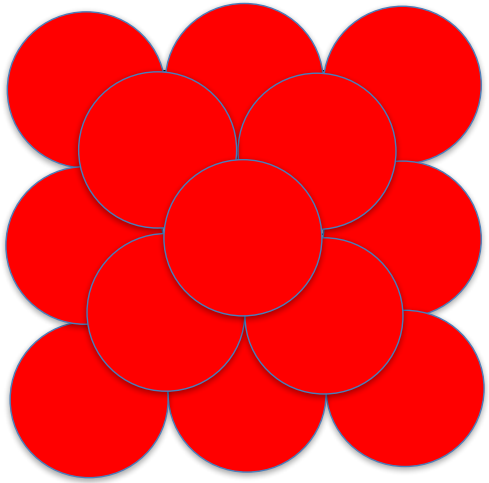
# Modeling disasters







1 square -> 5 locations



4 square -> 13 locations

# of possible disaster zones: (w * h) + ((w+1) * h+1)
Width = w
Height = h

25 X 12 = 300 squares
**668 minus 0-1 element gets affected.**

# Algorithm

Many components of the problem is NP-Hard.

- CPP reduces to a Facility Location Problem and is proved to be **NP Hard.**
- VNE is NP-Hard.

**Decomposition technique based heuristic algorithm:** to reduce the computational complexity

The main idea of our algorithms is to decompose the primal problem into |R| sub-problems and solve these sub-problems separately.

For VNE: this means, the mapping for virtual nodes and links are completed in ordered phases.

1. For each node, find the set of nodes within reachability circle.

2. Find list of minimum nodes that must exist in the VN, considering each switch need 2 closeby controllers, considering capacity requirements. Minimal set cover. Sort those list acc. to number nodes that must be in the VN.

3. Here, we have list of nodes that satisfy initial latency requirement and the capacity requirement. Switch assignment is also done at this stage for all options.

4. Among those lists, calculate worst case path setup. Find farmost node pair. Calculate worst case path setup. We have switch assignments. Shortest paths NodeA to contA + nodeB to cont b + contb to cont a -> are considered worst case. Eliminate lists that do not meet worst case path setup requirement. Sort the rest.

5. During the eliminations if no lists meet requirements, add more nodes in prev. steps.

**From now on, consider disaster resilience.**

Sort the remaining lists acc. to maximum damage done based on the affected number of elements/connections (s-c shortest paths + controllers + c-c connections(connect all controllers to the closest controller to them with shortest path)) by a certain-sized disaster.

Up until now, we decide on the number of nodes, their locations, switch assignments.

**Deciding on VN links and mappings:**

2-connected.

Ensure connectivity in case of any disaster sized r.

Control plane latency requirement for a full synchronization.

Minimize # of controllers by increasing controller number only when it is needed.
Not every distribution of dcs or amount gives feasible solutions.
Only consider the ones that do give.

# Sensitivity analysis

- Effect of DC locations and amount.
- The effect of topology? # of nodes, average link lengths, and connectivity properties. If links are longer, more controllers needed when the delay tolerance is same.
- The effect delay tolerance?
- The effect of the size of the disaster?
- The effect of controller capacity?