Interactive Visualisation System for Wide Area Network Paths

Team 2: Ibrahim, Biwen, Emmanuel, Farzad, and Wei

Background

- Cloud services are being deployed on third party cloud services and are being used on a daily basis
- Quality of service (QoS) of these services are latency sensitive and negatively perceived by users
- Network latency can have effect wide area network (WAN)

Problem

- Network latency in wide area networks (WAN) have tremendous effect on the performance and responsiveness of the service .

Hypothesis

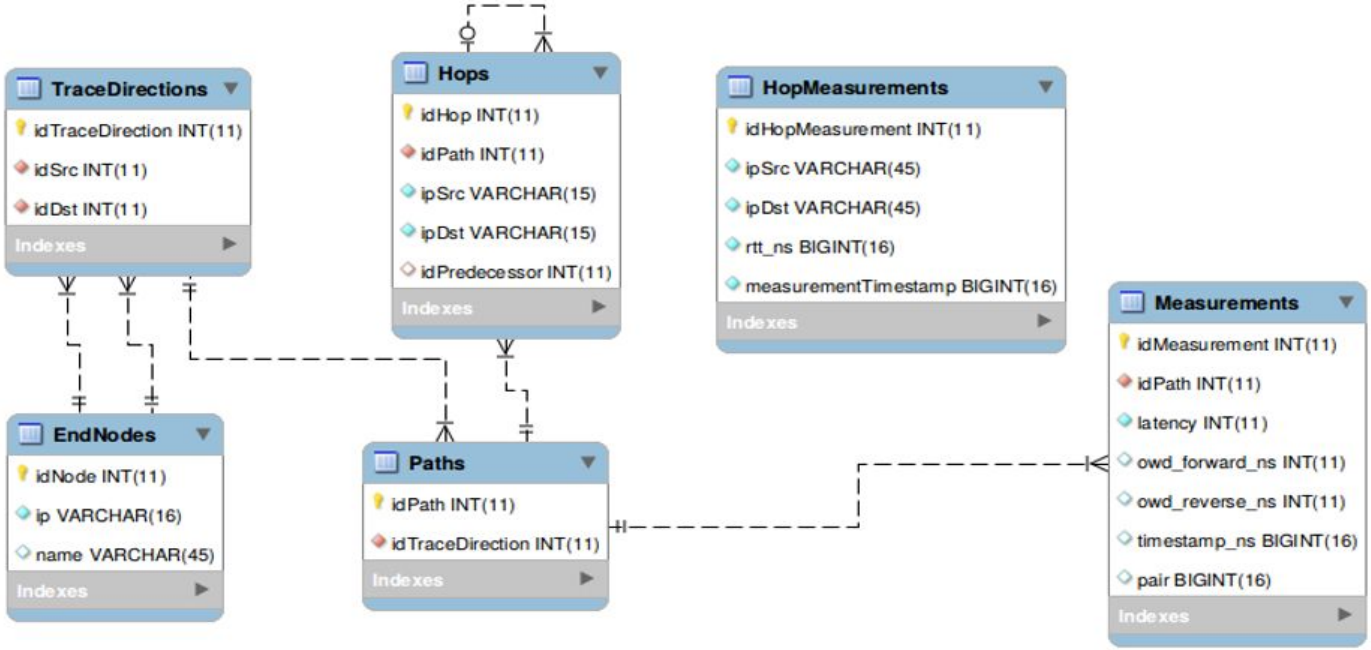
- When using existing topologies we can propose a new topology that can reduce network latency.
- Is it possible to design a system to improve the existing network topologies using offline data to improve current topology.

Goals

- Analysing the provided dataset network paths, extending it with geographical information and analyse the latency.
- Design and develop a visualization tool that will show the results from the database on the world map.
- Design and develop expert infrastructure change system (EICS)

Geo-Distributed Applications

MySQL database



EndNodes

idNode	ip	name
1	34.253.136.165	Ireland
2	13.113.103.115	Tokyo
3	34.211.167.25	Oregon
4	13.124.159.69	Seoul
5	34.229.118.49	N.Virginia

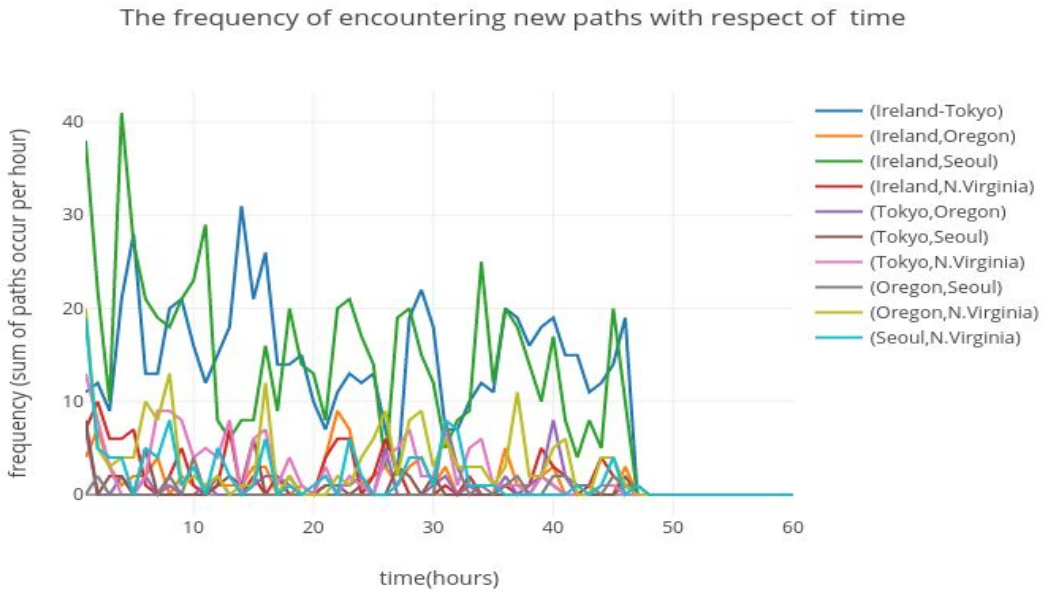
TraceDirections

idTraceDirection	idSrc	idDst
1	1	2
2	1	3
3	1	4
4	1	5

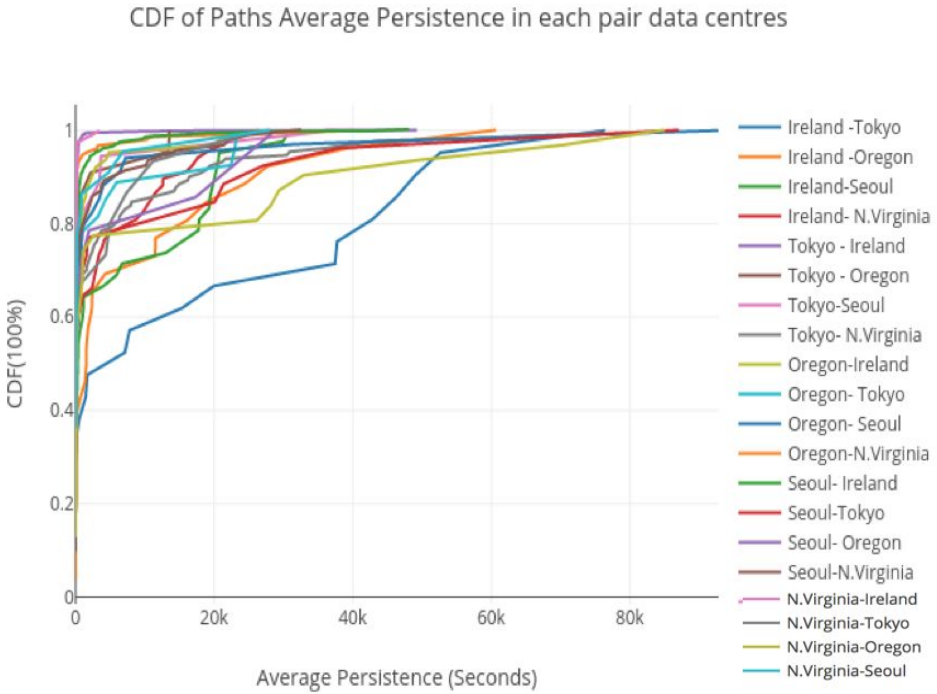
GeoIp

idNode	ip	name
1	34.253.136.165	Ireland
2	13.113.103.115	Tokyo
3	34.211.167.25	Oregon
4	13.124.159.69	Seoul
5	34.229.118.49	N.Virginia

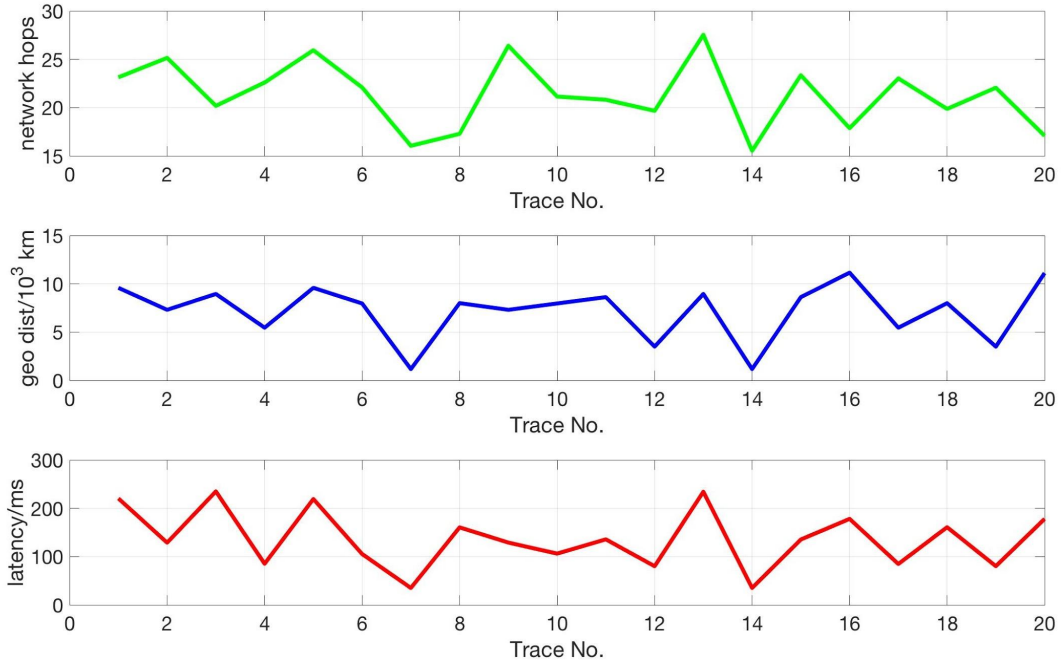
Demonstrate that the number of network paths between any two datacenters is finite



Analyze path persistency



Correlate network latency with geographic distance



1 Ireland-Tokyo
2 Ireland-Oregon
3 Ireland-Seoul
4 Ireland-N.Virginia
5 Tokyo-Ireland
6 Tokyo-Oregon
7 Tokyo-Seoul
8 Tokyo-N.Virginia
9 Oregon-Ireland
10 Oregon-Tokyo
11 Oregon-Seoul
12 Oregon- N.Virginia
13 Seoul-Ireland
14 Seoul-Tokyo
15 Seoul-Oregon
16 Seoul-N.Virginia
17 N.Virginia-Ireland
18 N.Virginia-Oregon
19 N.Virginia-Seoul
20 N.Virginia-N.Virginia

Goals

- Analysing the provided dataset network paths, extending it with geographical information and analyse the latency.
- **Design and develop a visualization tool that will show the results from the database on the world map.**
- Design and develop expert infrastructure change system (EICS)

Research for visualization tools

- A literature study has been conducted for finding a suitable visualisation tool and GeoIP service.
- Three visualization tools have been analyzed thoroughly, these tools are : Gephi, MapBox and Google Maps API.
- Google Maps API was chosen as visualisation tool since it meets all the requirements and functionality for project completion.

MaxMind GeoIp services

- MaxMind was used to extend the database with geographic information

Positives :

- Easy to extend to the existing database
- Data format can fit any database , no database schemas
- Good online documentation

OBS!!

- A problem was discovered when translating IP addresses using MaxMind.

Drawing the topologies

- Used an open-source Internet Topology Zoo
- Converted GML-formatted topologies to JSON
- Used JavaScript to display JSON-formatted topologies on the world map

Google Maps API

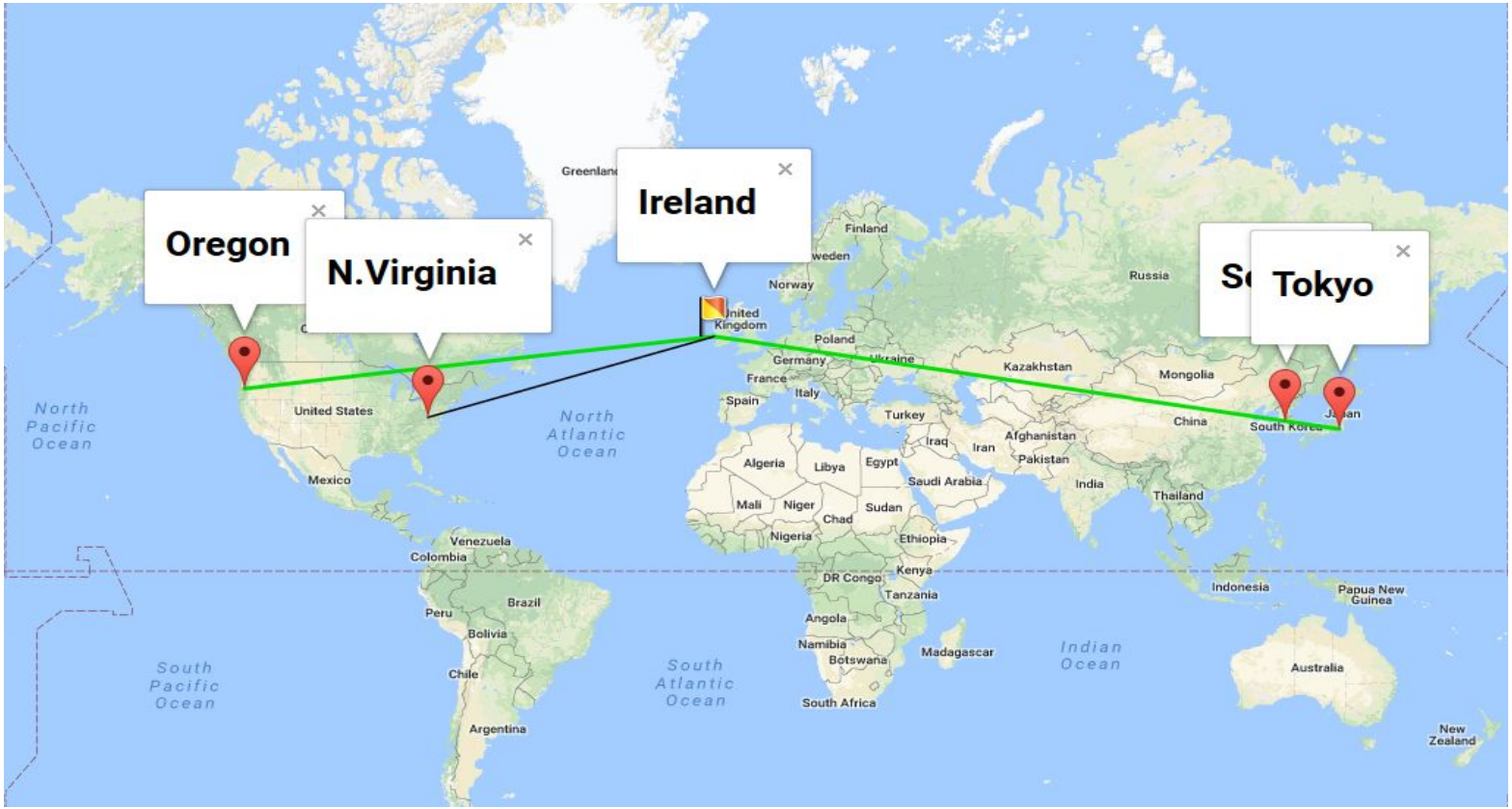
Google Maps API

Advantages:

- Suitable for web browsers since it is web based
- Easy to develop compared to other visualisation tools
- Accurate real time information for mapping, plotting and locating.

Disadvantages:

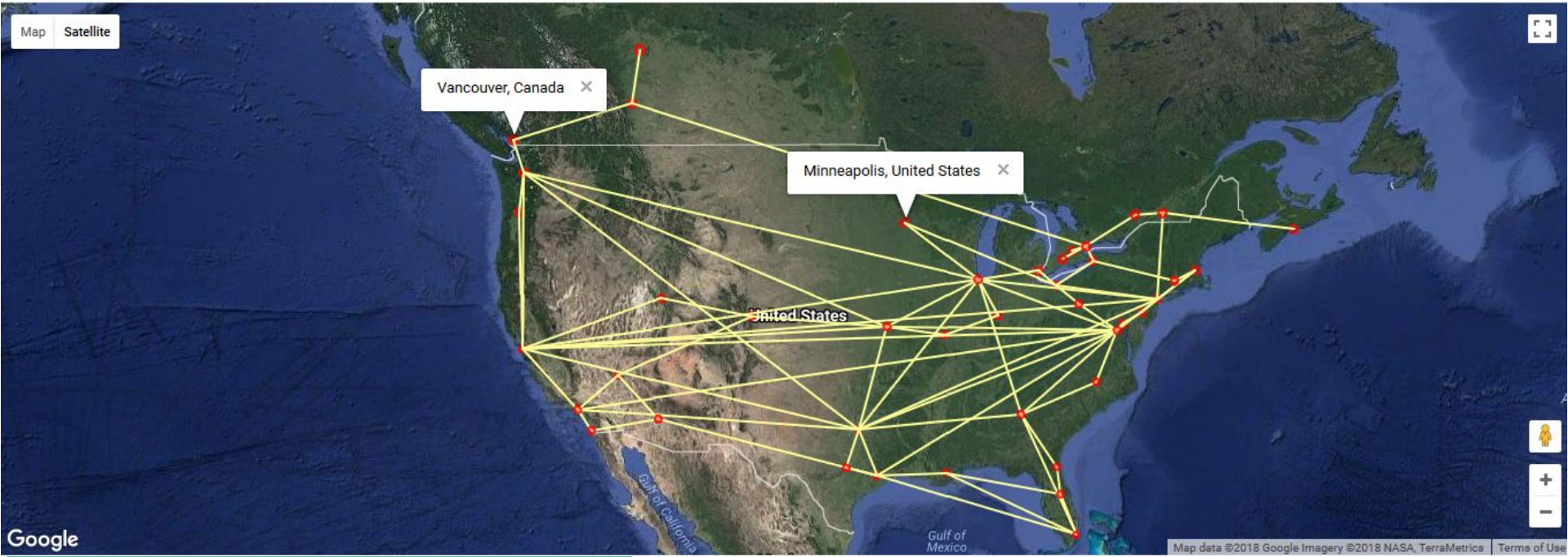
- Does not afford systematic IDE
- Some features are missing


Oregon
N.Virginia
Ireland
Tokyo
North Pacific Ocean
United States
Mexico
North Atlantic Ocean
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Greenland
Finland
Norway
United Kingdom
Poland
Germany
France
Spain
Italy
Russia
Kazakhstan
Mongolia
China
South Korea
Turkey
Iraq
Iran
Afghanistan
Pakistan
India
Thailand
Indonesia
Papua New Guinea
Australia
New Zealand
Algeria
Libya
Egypt
Saudi Arabia
Mali
Niger
Chad
Sudan
Nigeria
Ethiopia
Kenya
DR Congo
Tanzania
Angola
Namibia
Botswana
Madagascar
South Africa
Venezuela
Colombia
Peru
Brazil
Bolivia
Chile
Argentina

Design and develop an Interactive Visualization Tool

- Read network paths information from database(need to be changed)
- Construct topology graph and display it on the world map
- Place the nodes (data centers) on the map and make paths between them
- Create a simple UI with drop down menus to select between two different data centers.

Interactive Visualization System



Zoom in on: ◉ World ○ USA ○ China ○ Germany ○ France ○ Finland

Shortest path:
Source node: Vancouver, Canada Destination node: Minneapolis, United State Request shortest path Reset

New topology: Choose graph: USA Maximum cost: 700 Improve topology

Goals

- Analysing the provided dataset network paths, extending it with geographical information and analyse the latency.
- Design and develop a visualization tool that will show the results from the database on the world map.
- **Design and develop expert infrastructure change system (EICS)**

Design and develop the expert infrastructure change system

- Develop an expert infrastructure change system (EICS) for proposing alternative routes on the graph
- Integrate EICS with IVS
- Evaluate the performance of the EICS through testing

Rules

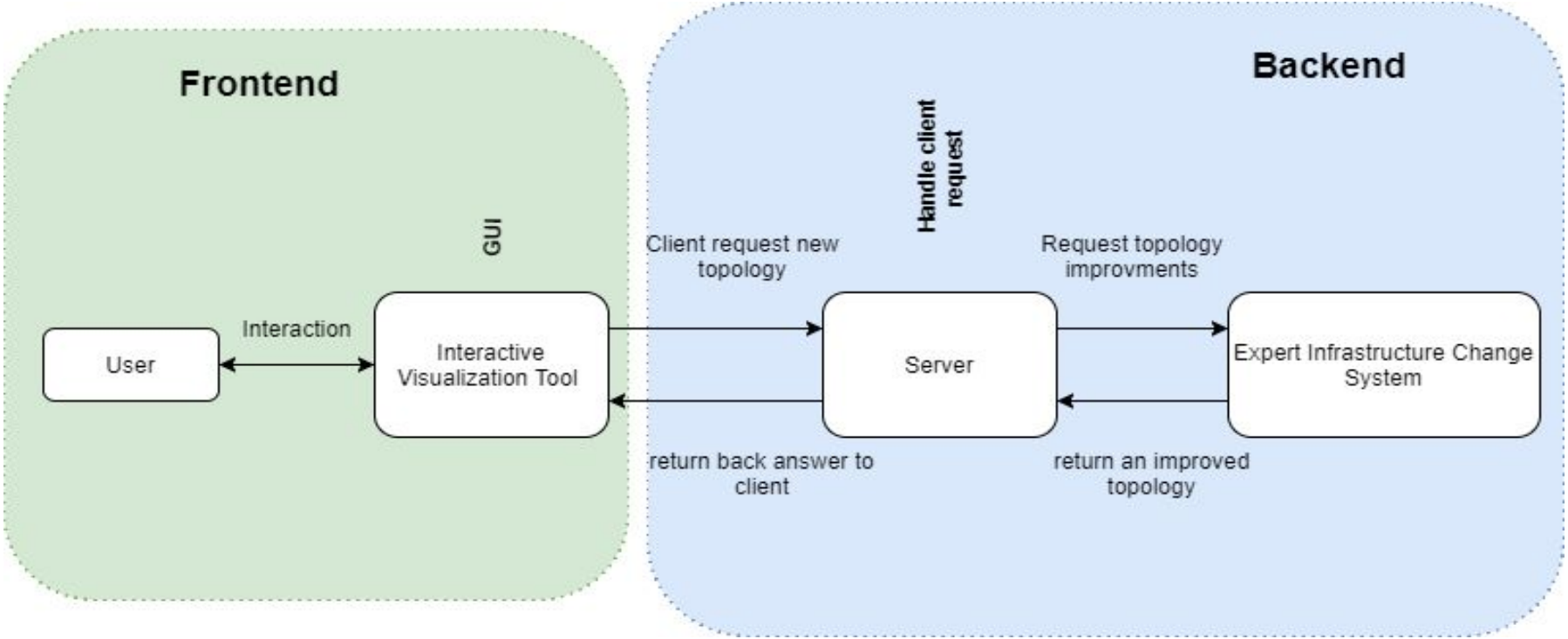
- R1. It is possible to place additional vertices (network hops) anywhere on the world map and use these points to connect any number of existing neighbouring vertices.

- R2. Network latency is proportional to the geographic distance as estimated earlier.

Add one more slide here

- Target: algorthm
- Test for performance
- Goal and budget

Workflow of our EICS



Calculating distance between geographic nodes

- The calculation is using the latitudes and longitudes of two points
- Converting latitudes from degrees to radians
- Subtracting longitudes and converting the result to radians
- Use the spherical law of cosines formula:

$$d = \operatorname{acos}(\sin \varphi 1 \cdot \sin \varphi 2 + \cos \varphi 1 \cdot \cos \varphi 2 \cdot \cos \Delta\lambda)$$

- $\Phi 1$ and $\varphi 2$ are the latitudes of the first point and second point while $\Delta\lambda$ is the difference between the longitudes of the two points.

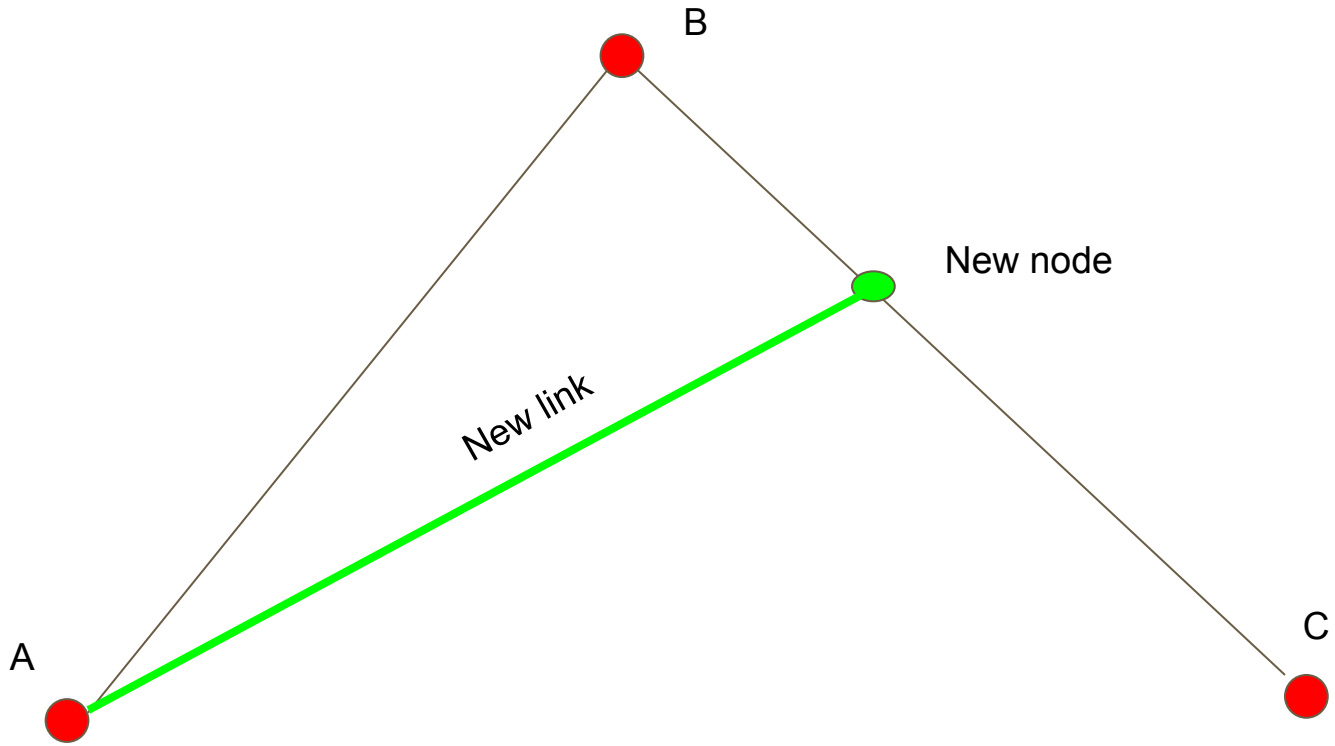
Proposed Algorithms

1. Random Triangle Zone

2. Draw circle to build constraint

3. Brute force algorithm

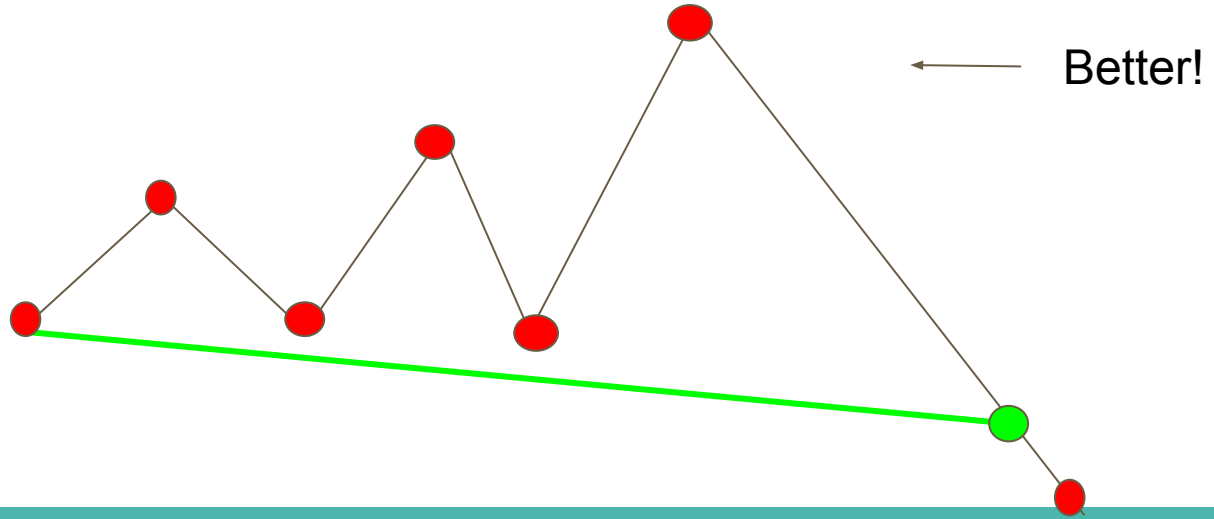
Random Triangle Zone



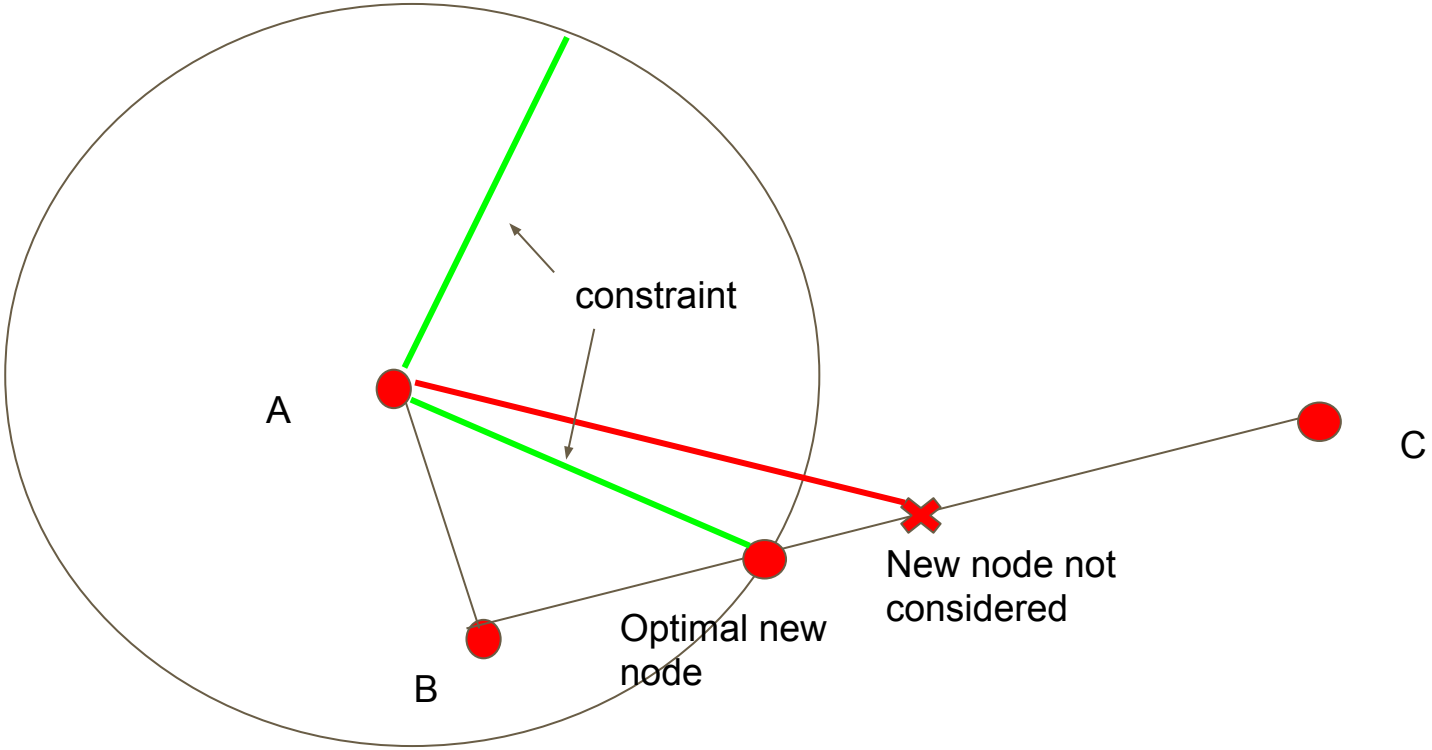
Random Triangle Zone

Difficulty and limits.

The reduced distance (latency) is limited.



Draw circle to build constraint

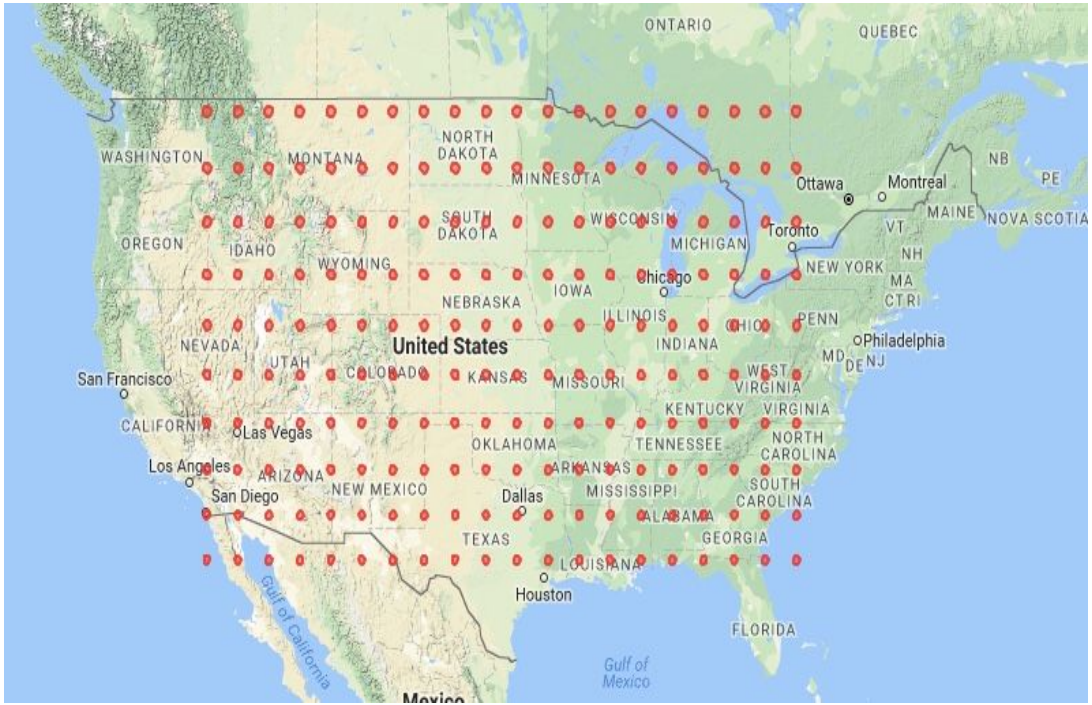


Brute force algorithm

Generate a grid of nodes with fixed distance in between.

Link them with existing nodes and test for Dijkstra algorithm again.

Find the optimized node that shortens the total distance



Brute force algorithm Pseudo code

```
Run Dijkstra's algorithm for specified source node
Store the cost to the target as the best distance
For all new nodes
        Add node Nx to the Graph
        For all pair of nodes that existed in the initial graph
          Add the two new nodes as neighbors of Nx if constraint is met
          Run Dijkstra's algorithm for the specified source node
          Get the distance to the target node (cost of the target)
          Remember this node and its neighbors if the distance is
        better.
```

Brute force search

Disadvantages and limits

Complexity is proportional to the number of route in the graph, which leads to many practical problems:

1. The running time tends to grow very quickly at the same time with the size of the graph increases.
2. The density of
3. Exclude the points

Limits of our EICS

- Brute-force searching algorithm does solve the problem but is not so effective.
- The EICS is not able to learn from mistakes (e.g. geographical barriers), no common sense used in making decisions.
- We do not consider balancing network traffic (it should be considered in real life if we sale EICS to network providers).

Evaluate session

Future work

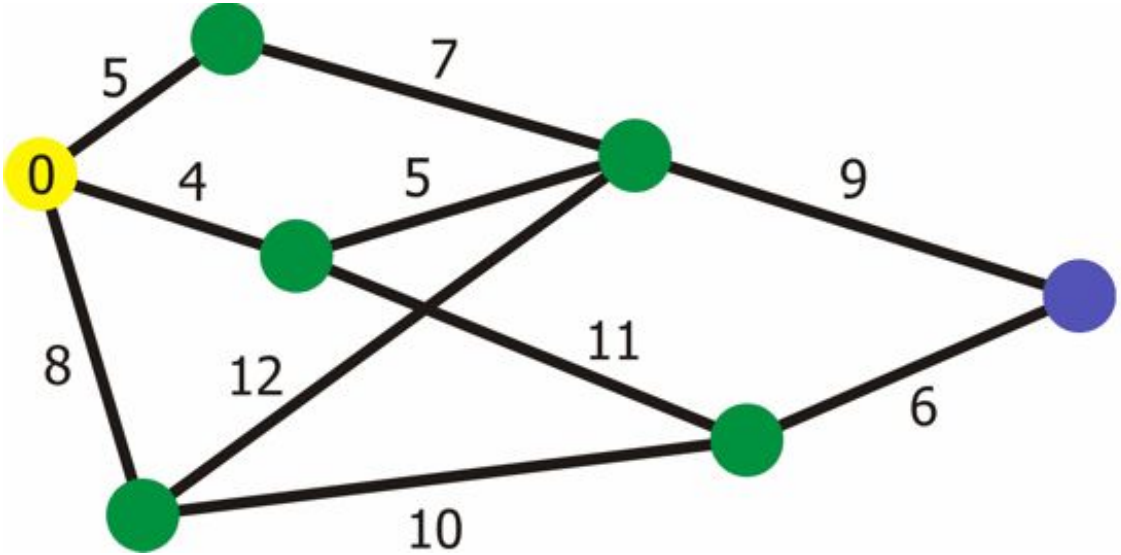
- Find better algorithm to reduce the total distance (latency) more efficiently.
- Employ load balance to enhance reliability and increase resource utilization.
- Use Machine learning to learn geographical barriers and avoid it.

Demo time

Any Questions?

Running Dijkstra's algorithm

- Dijkstra is run on the server-side
- Calculate the distance between each node, then run Dijkstra



Dataset sample

```
"nodes": [
{
"Latitude": 36.17497,
"Country": "United States",
"Internal": 1,
"id": "Las Vegas",
"num": 1,
"Longitude": -115.13722
},
...]
```

```
"links": [
{
"LinkType": "DS-3",
"source": "Las Vegas",
"snum":1,
"LinkLabel": "45 Mbps DS-3",
"LinkNote": "45 Mbps ",
"target": "Phoenix",
"tnum":12
},
...]
```