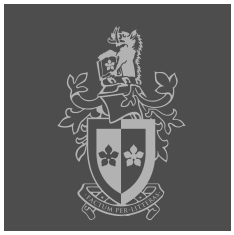
SWINBURNE UNIVERSITY OF TECHNOLOGY

Re-imagining Vivaldi with HTML5/WebGL

PhD Candidate: Djuro Mirkovic
Supervisors: Prof. Grenville Armitage and Dr. Philip Branch

dmirkovic@swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology



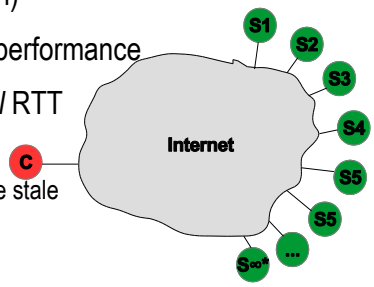
Outline

- Problem Scenario
- Motivation
- Goal
- Network Coordinate System (NCS)
- Model Space
- Vivaldi Discussion and Examples
- Demo
- Summary

Problem Scenario

Client wants to select a Server with the lowest round trip time (RTT)

- Measure RTT with all Servers?
  - Wastes resources (eg. Bandwidth)
  - May overflow NAT Table – poor performance
  - May take a long time to gather *all* RTT measurements
    - After time *t*, RTT measurements are stale



Motivation

- Peer-to-peer (P2P) applications treat the Internet as a 'black box'
- Client-Server model optimal selection can be improved
- Scenarios: Network Coordinate System (NCS) with large number of nodes or small data size can improve performance and resource allocation

Goal

A model that is able to predict the distance between two Nodes, given only a few measurements.

Network Coordinate Systems (NCS)

- Centralised
  - Predict using Synthetic Coordinates as described in Global Network Positioning (GNP) [1]
  - Requires “Landmark(s)” (most likely being Server Nodes or accurate Nodes)
- Decentralised
  - Predict using Network Coordinate System as described in Vivaldi [2]
  - “Landmark(s)” are optional

[1] - http://www.cs.rice.edu/~eugeneng/papers/INFOCOM02.pdf
[2] - http://conferences.sigcomm.org/sigcomm/2004/papers/p426-dabek111111.pdf

Model Space

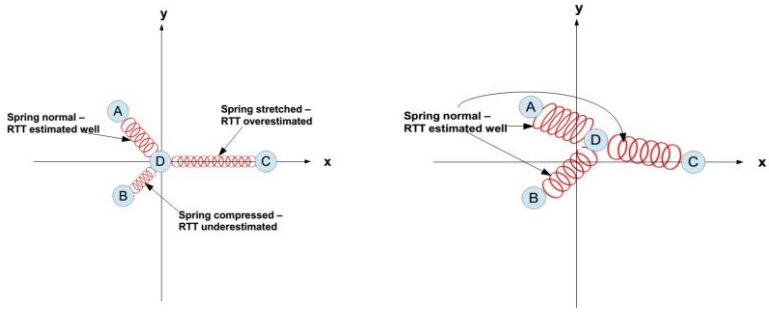
- N-dimension Euclidean Model
  - At least 2D Model is required
  - Increasing *N*-dimensions substantially in Euclidean Model does not have significant improvements
    - May have an opposite undesired effect – increasing the computational power required for higher *N*-dimensions
  - Using a 3D to 5D Model is adequate
- 2D + Height Vector Model
- Spherical Model
- Cylindrical Model

Decentralised - Vivaldi

- Vivaldi is solving a simultaneous equation distributed in time and space
- Nodes *iteratively* update their own coordinates based on current network state
- A “Local Node” that is communicating with “Remote Node(s)” requires a very *small sample* of “Remote Node(s)” to get an overall state
- Nodes communicating can ‘piggyback’ on existing application traffic
  - Adds very little overhead

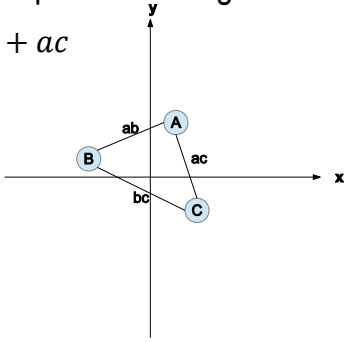
Springs Overestimate and Underestimate RTT

- Difference between **measured** and **calculated** RTT
- "Springs" compress and stretch to move Node D to new coordinate space after *n* iterations of Vivaldi
  - Push and Pull effect (Hooke's Law)



Triangle Inequality

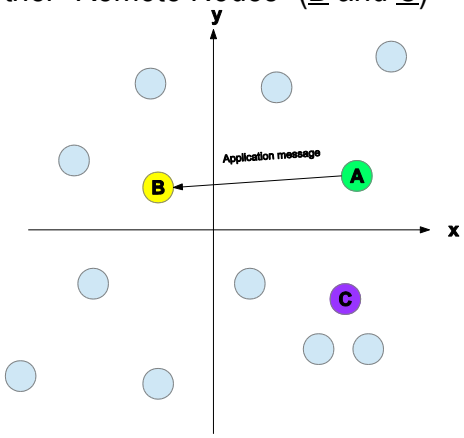
- NCS assumption is that Triangle Inequality holds for Node paths.
- "The sum of the lengths of any two sides, must be greater than or equal to the length of remaining side." [1]
  - Eg. $bc \leq ab + ac$



[1] -https://en.wikipedia.org/wiki/Triangle_inequality

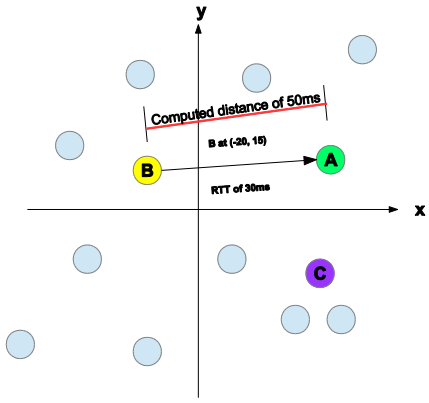
Vivaldi Example 1 (Step 1)

- "Local Node" A will do sequence of communication with other "Remote Nodes" (B and C)



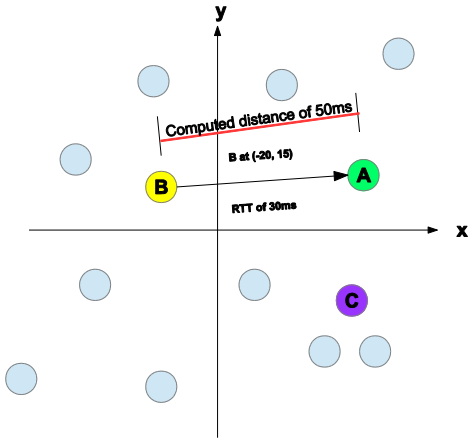
Vivaldi Example 1 (Step 2)

- A receives B's Coordinates (-20, 15) and RTT (30ms)
- A calculates distance to B (50ms) $RTT(A,B) = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$



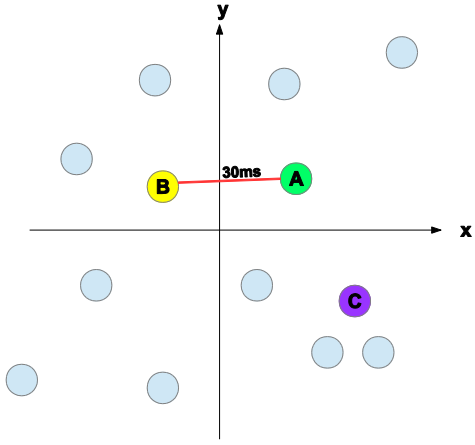
Vivaldi Example 1 (Step 3)

- **Overestimated**: A needs to be pulled towards B



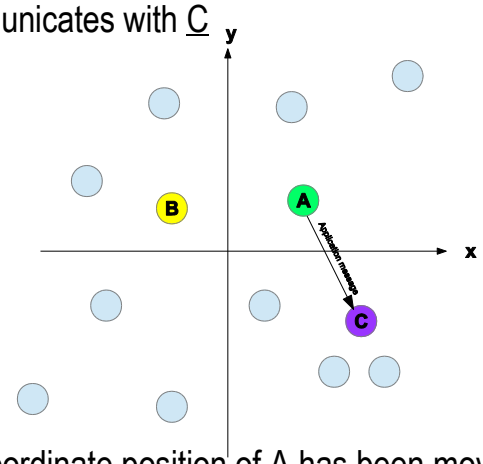
Vivaldi Example 1 (Step 4)

- A (over *n*-iterations – *not instantly!*) approaches B's **measured** RTT. A adjusts coordinates (A_x, A_y)



Vivaldi Example 1 (Step 5)

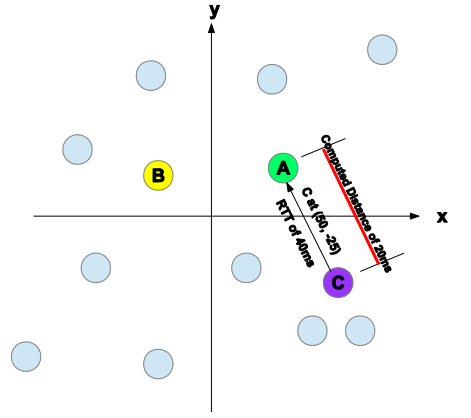
- A communicates with C



- Note: coordinate position of A has been moved (when A completed one iteration of Vivaldi with B)

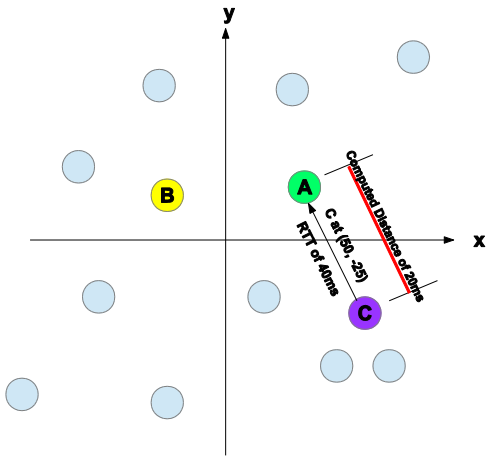
Vivaldi Example 1 (Step 6)

- A receives C's Coordinates (50, -25) and RTT (40ms)
- A calculates distance to C (20ms) $RTT(A,B) = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$



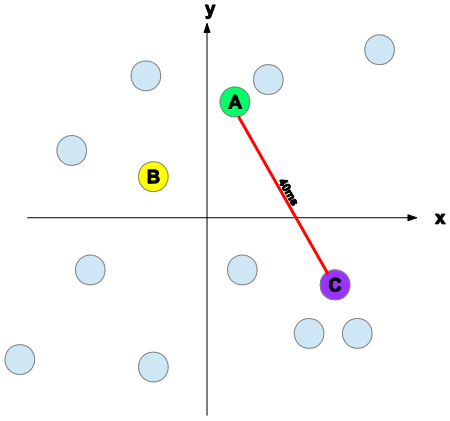
Vivaldi Example 1 (Step 7)

- **Underestimated**: A needs to be pushed away from C



Vivaldi Example 1 (Step 8)

- A (over *n*-iterations) approaches C's **measured** RTT. A adjusts coordinates (A_x, A_y)

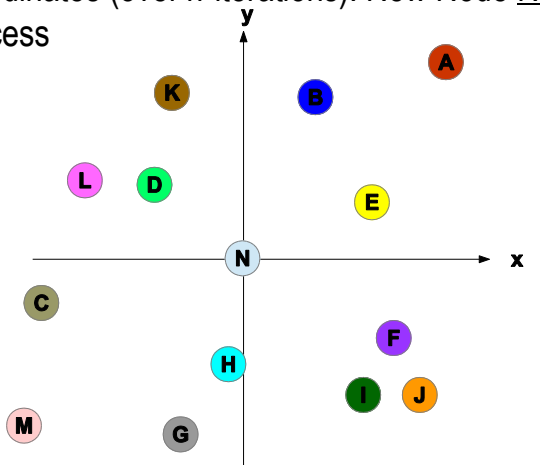


Vivaldi Example 1 (Step 9)

- Previous slide, the distance between A and B now would be wrong
  - Vivaldi process will *iterate* until all Local and Remote Nodes ***measured*** RTT have same ***calculated*** distance (or RTT)

Vivaldi "Landmarks" Example 2 (Step 1)

- 13 Stable "Landmark" Nodes (A to M) calculated their coordinates (over *n*-iterations). New Node N joins the process

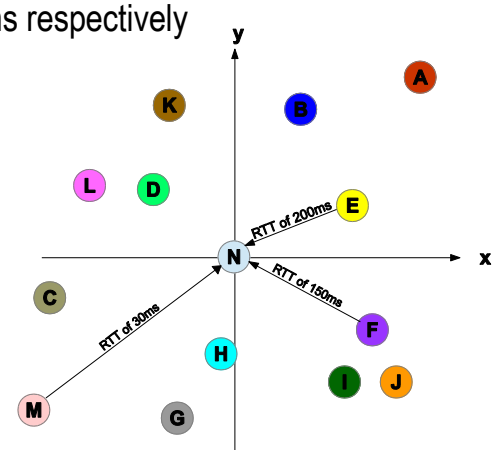


Vivaldi "Landmarks" Example 2 (Step 2)

- 2D Euclidean Model requires at *least* 3 Remote Node samples, otherwise many solutions may exist in NCS
  - Number of Remote Node samples is equal to number of dimensions + 1 (3D would need 4, 4D would need 5, etc)
- N needs 3 Remote Node samples:
  - Does N select 3 random?
  - Does N select more than 3 random?
  - If "Landmarks" used:
    - Does N select 3 shortest RTT **measured**?
    - Does N select 3 longest RTT **measured**?
    - Does N select 3 different geographical locations?

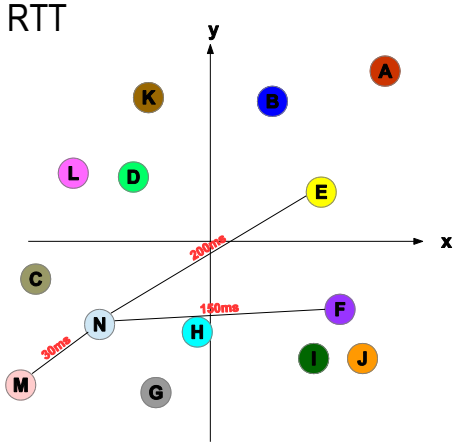
Vivaldi "Landmarks" Example 2 (Step 3)

- N *randomly* selected Remote Nodes (E, F, and M). **Measured** RTT from E, F, and M are 200ms, 150ms, and 30ms respectively



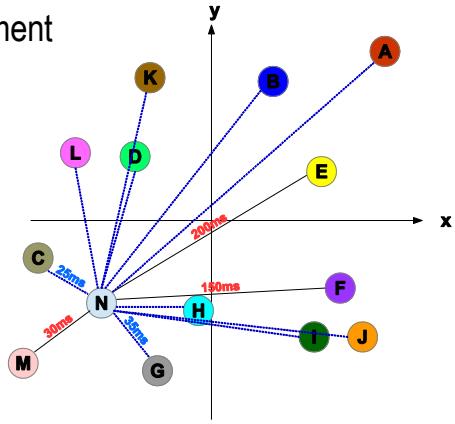
Vivaldi "Landmarks" Example 2 (Step 4)

- N after *n*-iterations of Vivaldi determines coordinates and satisfies the computed RTT to be the same as measured RTT



Vivaldi "Landmarks" Example 2 (Step 5)

- RTT between N and Remote Nodes (A, B, C, D, G, H, I, J, K, and L) can be predicted by the Euclidean Distance between their coordinates **without** direct measurement



Vivaldi "Landmarks" Example 2 (Step 6)

- N best option is C (25ms)
- N calculates the predicted RTT for A, B, C, D, G, H, I, J, K, and L faster than it takes to measure the RTT for each Remote Node
- Previous scenario had only 13 Remote Nodes, what if we had a large number of Remote Nodes (10,000, 100,000, 1 million, etc)?

Vivaldi space rotation

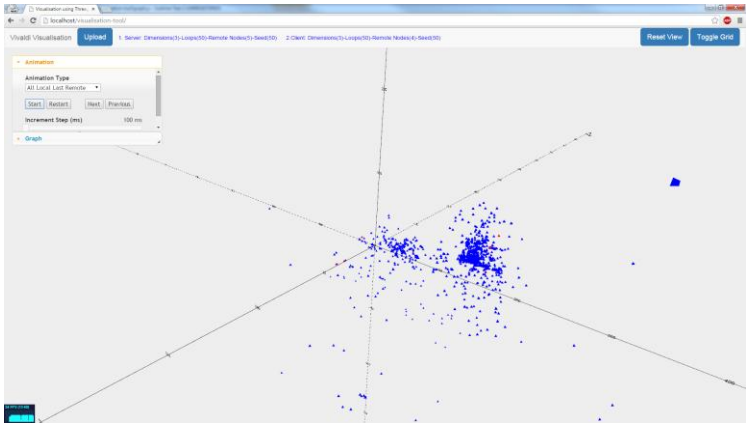
- Each Node in the Vivaldi process starts at the origin, and obtains a random starting position
- Vivaldi with different *seeds* (varying random starting position) using the same experimental data, an interesting observation is that:
  - the overall stable position in NCS in one *seed* universe becomes nearly like a rotation in the second *seed* universe

Vivaldi Caveats

- Triangle Inequality violations
- Malicious Nodes can affect the calculated coordinates

Demo of 3D Vivaldi Simulations

- Hard to conceptualise the overall Vivaldi process without a suitable simulation tool



Summary

- Vivaldi is a decentralised NCS
- 3D – 5D Euclidean Model is an acceptable model to use in NCS
- Vivaldi can optimise application performance and resource allocation

Acknowledgement

- Thank you very much to Prof. Grenville Armitage and Dr. Philip Branch for their supervision support throughout my PhD.

Questions?