Neighbor Selection Based on TIV Severity Sort Model in Vivaldi Network Coordinate System

Peng Sun, Yang Chen*, Beixing Deng, Xing Li

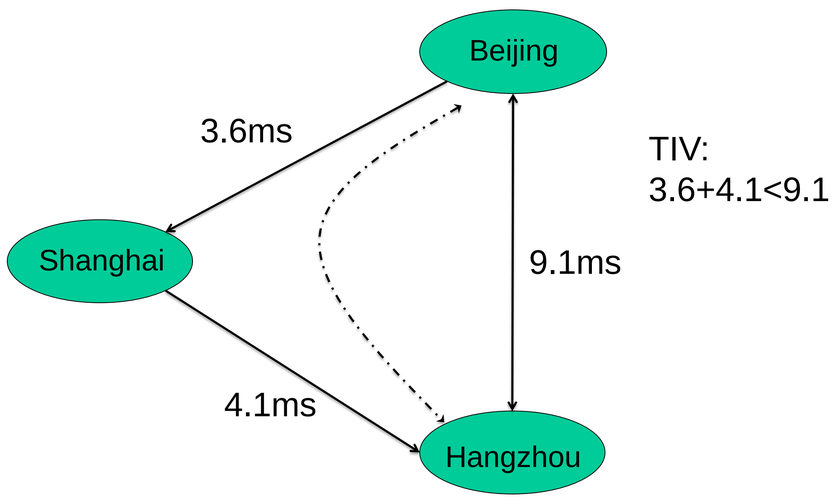
Department of Electronic Engineering, Tsinghua University, Beijing, China

*Institute of Computer Science, University of Göttingen, Germany

What TIV is

TIV is abbreviation for Triangle Inequality Violation.

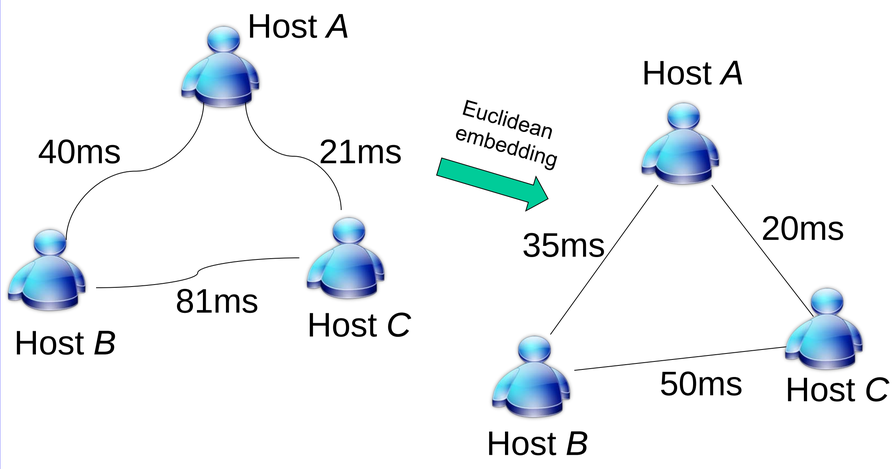
Here is a real example of TIV in Internet of China.



Traffic from Beijing via Shanghai to Hangzhou costs less time than traffic from Beijing directly to Hangzhou due to routing policies. It violates the triangle inequality of Euclidean space.

How TIV affects accuracy of NC

In Euclidean space embedding of Network Coordinate system, the edges who are in TIV are forced to shrink or stretch, which definitely introduces inaccuracy.



The more severe the TIV of a dataset, the higher the 90th percentile relative error (**NPRE**) of coordinate calculation.

Dataset	Triples in TIV	NPRE
AMP	4.29%	20.2%
King	12.32%	47.6%
P2PSim	17.10%	56.3%
PlanetLab	25.71%	78.8%

What TIV severity is

$$TS_A = \frac{1}{|S|}\sum_{C\in S}\left\{\frac{1}{|S|}\sum_{B\in\Gamma}\frac{d(A,C)}{d(A,B)+d(B,C)}\right\}$$

S is the set of all hosts and $A,B,C \in S$.

Γ is $\{X \mid d(A,X) + d(X,C) < d(A,C), X \in S\}$

How our mechanism works

Focusing on the most representative Network Coordinate system – Vivaldi, we propose a new mechanism of neighbor selection based on TIV Severity Sort Model to improve *Vivaldi's* accuracy, named as *TIV Severity Adjusted Vivaldi*.

Algorithm : *TIV Severity Adjusted Vivaldi*

1: Neighbor Set Up

WHEN $K < L$
  select all existing hosts as i's neighbors
  incorporate i into S
WHEN $K \geq L$
  WHILE $j \in S$
    calculate TS_j and sort
  select smallest-TS L hosts as i's neighbors
  incorporate i into S

2: Coordinate Iteration

WHILE $i \in S$
  WHILE j is i's neighbors
    $w = e_i/(e_i + e_j)$
    $e_s = |\|x_i - x_j\| - rtt_{ij}|/rtt_{ij}$
    $e_i = e_s \times c_e \times w + e_i \times (1 - c_e \times w)$
    $\delta = c_c \times w$
    $x_i = x_i + \delta \times (rtt_{ij} - \|x_i - x_j\|) \times u(x_i - x_j)$

K: Number of Existing Hosts, L: Number of Neighbors

S: Set of Existing Hosts

How the performance is

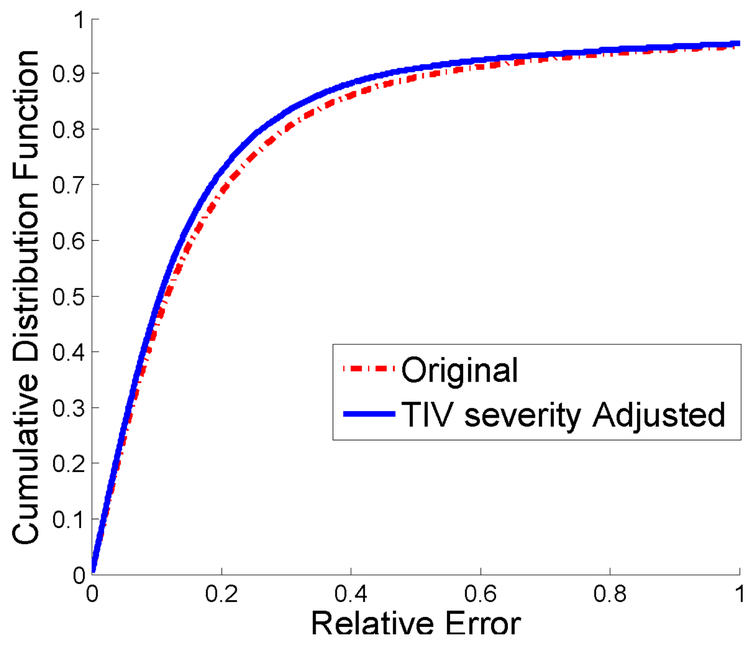


Fig.1 Cumulative Distribution Function of Relative Error

It is the comparison of the relative error of distance prediction.

The NPRE of our mechanism is 45.7%, while in the original Vivaldi it is 53.1%.

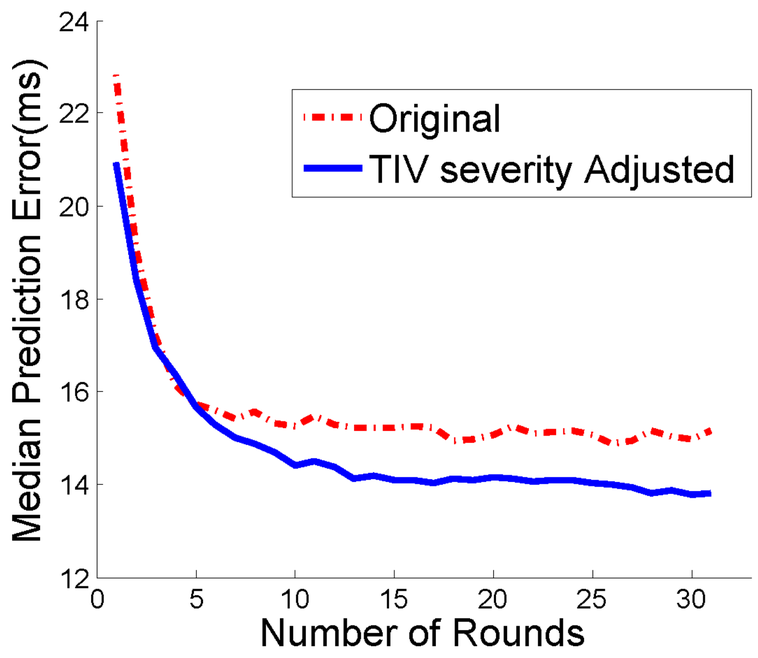
Fig.2 Convergence Behavior

It shows the evaluation of the numbers of rounds required for convergence under a flash-crowd scenario.

Our mechanism converges faster than the original one.

The final median prediction error of our mechanism is 13.78ms, compared to 14.96ms in the original Vivaldi.

NOTE: 256 nodes are randomly extracted out of 1143-node P2PSim for evaluation.



Conclusion

1. The 90th percentile relative error of Vivaldi is decreased by **13.9%** on the dataset of P2P-256.
2. The convergence rate is improved and the final median prediction error is **7.9%** smaller.

Acknowledgement

This work is supported by the Project Research on Method and Security of Network Coordinate Based Superspace-Subspace Embedding supported by NSFC (No.60850003).