Triangle Inequality Violation Avoidance in Internet Coordinate Systems

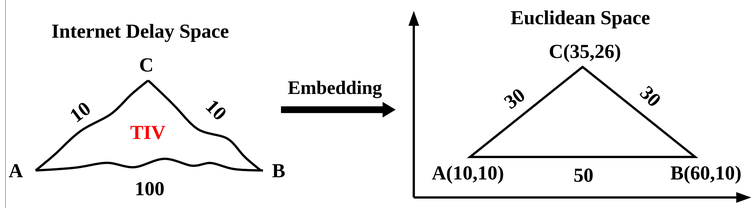
Yongjun Liao and Guy Leduc

University of Liège, Belgium

liao@montefiore.ulg.ac.be, Guy.Leduc@ulg.ac.be

Internet Coordinate Systems (ICS)

embed Internet delay space into a metric space so that the delay between two nodes is approximated by their distance in the embedded space.



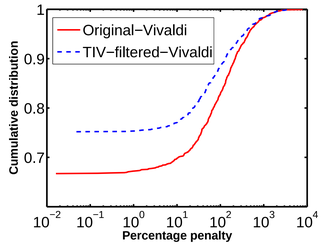
Triangle Inequality Violations (TIV)

$\overline{AB} > \overline{AC} + \overline{CB} \Rightarrow$ TIV

$\overline{AB} \Rightarrow$ TIV base

Impact of TIVs to ICS

The presence of TIVs causes large embedding errors which can misguide nearest neighbor selection and therefore hurt the performance of many Internet applications.



Nearest neighbor selection penalty (P2psim)

Our contributions

- **Learning to detect TIVs:** an effective TIV detection criterion is found by using a supervised learning method, namely decision trees.
- **TIV avoidance in ICS:** by iteratively excluding TIV edges from ICS based on the above learned criterion, the performance of neighbor selection is improved.

This work has been partially supported by the EU under project FP7-Fire ECODE.

Learning to detect TIVs

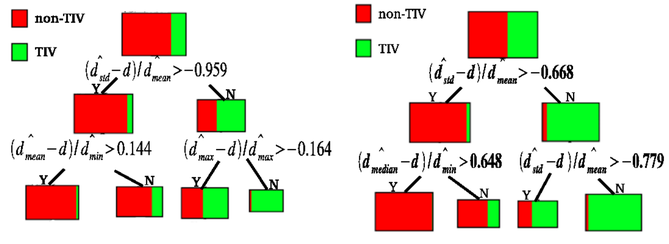
- collect training data from ICS algorithms such as Vivaldi
  - collection of inputs
    - measured delay: d
    - estimated delays: $\{\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_K\}$
    - statistics: $\hat{d}_{max}$, $\hat{d}_{min}$, $\hat{d}_{mean}$, $\hat{d}_{median}$ and $\hat{d}_{std}$
    - 64 input variables: $\frac{\hat{d}_{max}-\hat{d}_{min}}{d}$, $\frac{\hat{d}_{mean}-d}{\hat{d}_{max}}$, $\frac{\hat{d}_K}{d}$, $\frac{\hat{d}_{std}}{d}$, $\ldots$
  - collection of outputs
    - TIV: the longest edge on any TIV triangle.
    - non-TIV: otherwise.

	P2psim	Meridian
nodes	1740	2500
edges	$1740 \times 32 = 55680$	$2500 \times 32 = 80000$
TIV base proportion	23%	42%
input variables	64 per edge	64 per edge
output label	1 per edge	1 per edge

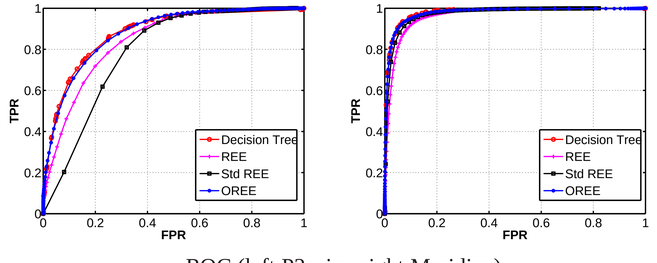
- decision trees are used to learn TIV classifiers
- a discriminative variable, $OREE$ ($\frac{\hat{d}_{std}-d}{\hat{d}_{mean}}$), is found

OREE: Oscillation and **R**elative **E**stimation **E**rror

- $\frac{\hat{d}_{std}}{\hat{d}_{mean}}$: relative oscillation measure
- $\frac{d}{\hat{d}_{mean}}$: relative error measure



Decision trees (left:P2psim, right:Meridian)



ROC (left:P2psim, right:Meridian)

TIV avoidance based on OREE

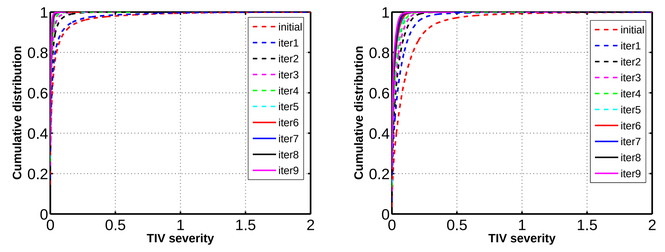
Idea: progressively exclude edges with small $OREE$ values as they are more likely to be TIVs.

- Vivaldi is started in a traditional manner with each node selects m random neighbors;
- After a period of time, each node probes another m random nodes and gets totally $2 \times m$ neighbor candidates.
- For each edge between a node and one of its $2 \times m$ neighbor candidates, the value of $OREE$ is computed.
- The neighbors of a node are updated by selecting the m candidates with large $OREE$ values and abandoning the others with small ones;
- This neighbor update procedure is repeated every T seconds.

Experiments and evaluations

- **TIV severity:**

$$\frac{\sum_{c \in S} d_{a,b}/(d_{a,c}+d_{c,b})}{|S|},\ if\ d_{a,b} > d_{a,c} + d_{c,b}.$$



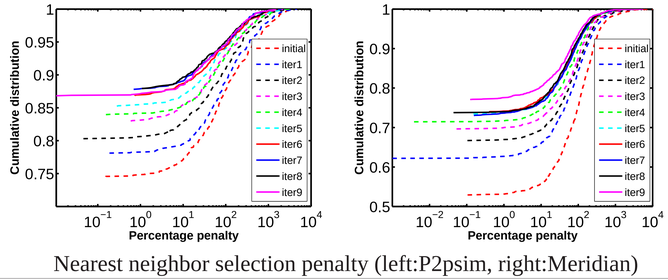
Distribution of TIV severity (left:P2psim, right:Meridian)

- **Nearest neighbor selection penalty**

A subset of the nodes are randomly selected as candidates for the nearest neighbor selection. For each node which is not in the candidate set, the nearest neighbor in the candidate set is detected in the original delay space and in the embedded space respectively.

$$\frac{(dist\_to\_selected - dist\_to\_optimal) * 100}{dist\_to\_optimal}$$

This metric attempts to answer the question ”how far is my nearest distance to a set of candidates?”



Nearest neighbor selection penalty (left:P2psim, right:Meridian)