TRAILS Mobility model

Implementation and Performance analysis

Leonardo Sarmiento

University of Bremen
Sustainable Communication Networks

18th September 2019

Supervised by:
Prof. Dr. Anna Förster
MSc Vishnupriya Kuppusamy
Dr.-Ing. Asanga Udugama

Universität Bremen

comnets

-Several researchers agree that a mobility model that does not capture the interaction of targeted users would give misleading results in the simulation of OppNets.

-Therefore, I present TRAILS as a fully scalable model capable of representing real scenarios in a reasonable manner.

TRAce-based ProbabILiStic Mobility Model

TRAILS is a mobility model in which a user performs movements extracted from recorded traces, but it chooses the destinations randomly

-TRAILS can be simulated with a larger number of users and for a longer time than the original traces. Therefore, TRAILS is a fully scalable and flexible mobility model.

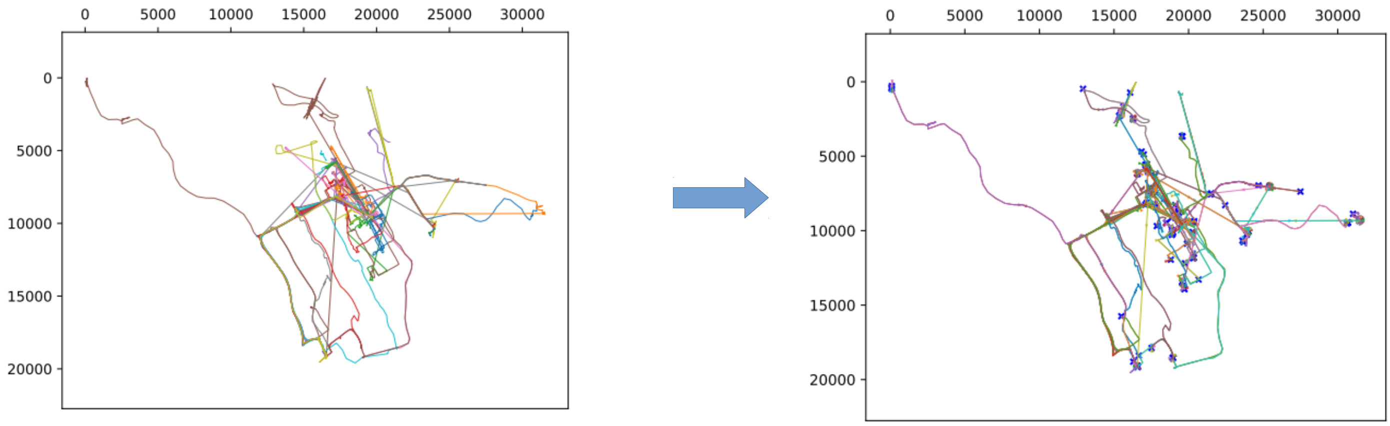
TRAILS requirements

- Traces from a real scenario
- Graph generator
- Model Simulator

3

-To be able to use TRAILS we need a trace-set, a graph generator, and a model simulator.
-In this presentation I am going to explain how the generator and the simulator work.
-Additionally I am going to present results of simulating traces of 4 different real scenarios and their corresponding TRAILS graphs.
-Finally I will present a recommendation to improve the performance of TRAILS

Graph generator

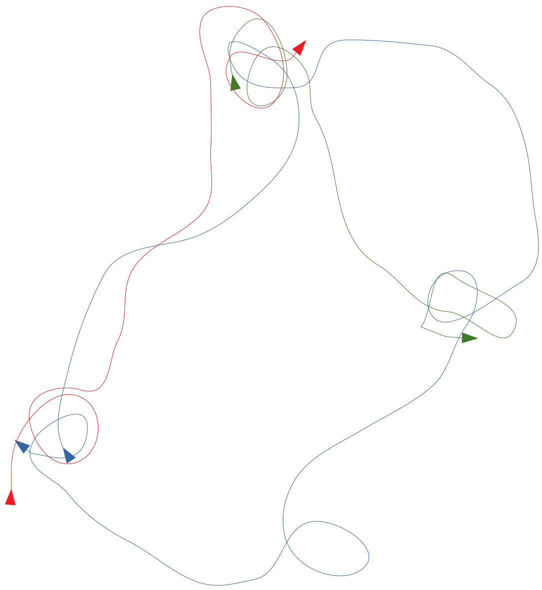


-A user in TRAILS moves to a random destination, It stays in that destination a random time and then it moves to another random destination.

-The destinations are called POI and the trajectories followed by a user to move between destinations are called Links.

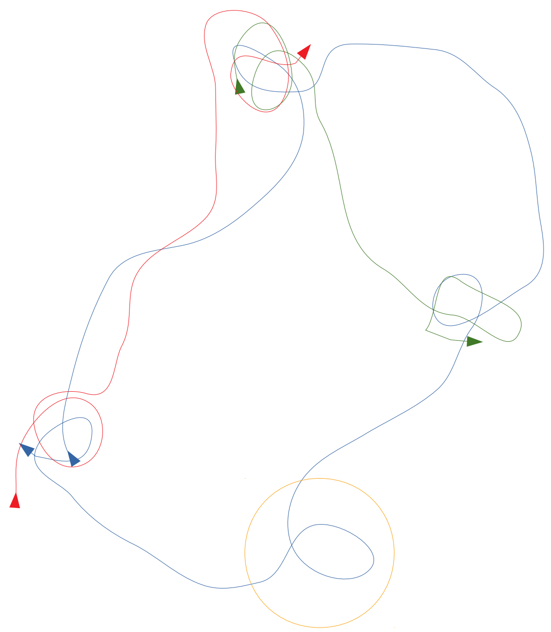
-POIs and Links are extracted from real traces with a graph Generator.

Find Points of Interest



-In this figure we have a trace-set of 3 users.
-And we need to find the POIs
-A POI is a place in which the users spend most of their time
-A POI is limited by a fixed diameter for example 30m, and it has a list of time intervals in which each time is longer than threshold for example 5 minutes
-The next slides describe the process used to find POIs.

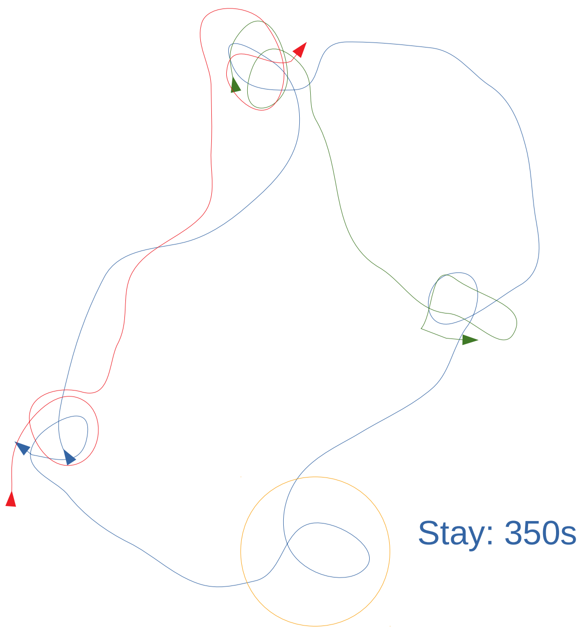
Algorithm description:



6

-First, by using the smallest enclosing circle algorithm proposed by Emo Welz in his paper “*Smallest enclosing disks*”, we search a set of consecutive trace-points enclosed by a circle smaller than a threshold. For example 30 meters

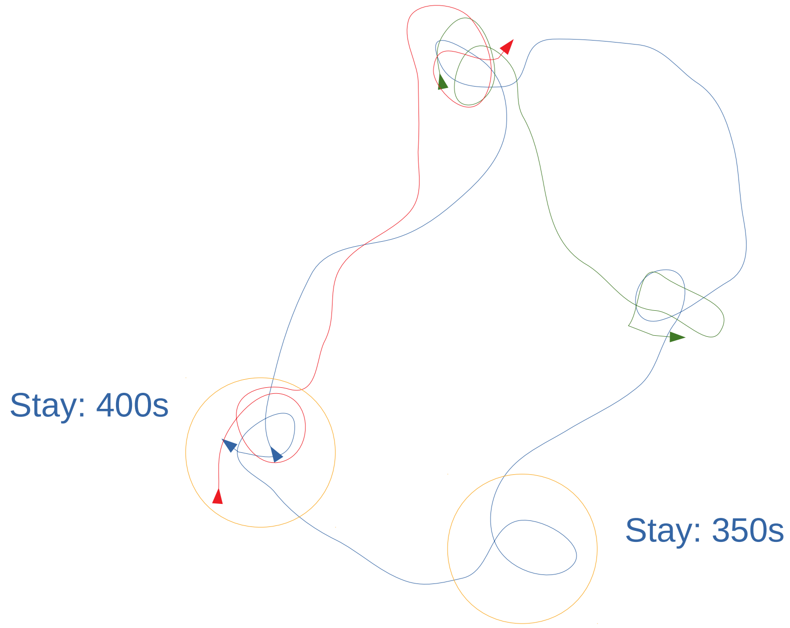
Algorithm description:



-We calculate the time that the user stayed inside the enclosing circle.

-If the time is longer than a threshold, for example 5 minutes, we create a POI and we add the time to a list.

Algorithm description:

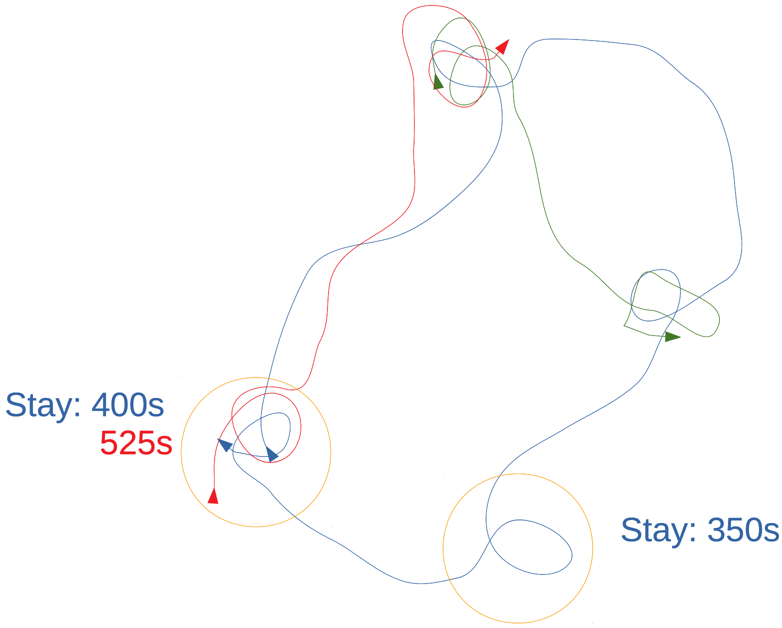


8

-We repeat the process until we find a new POI.

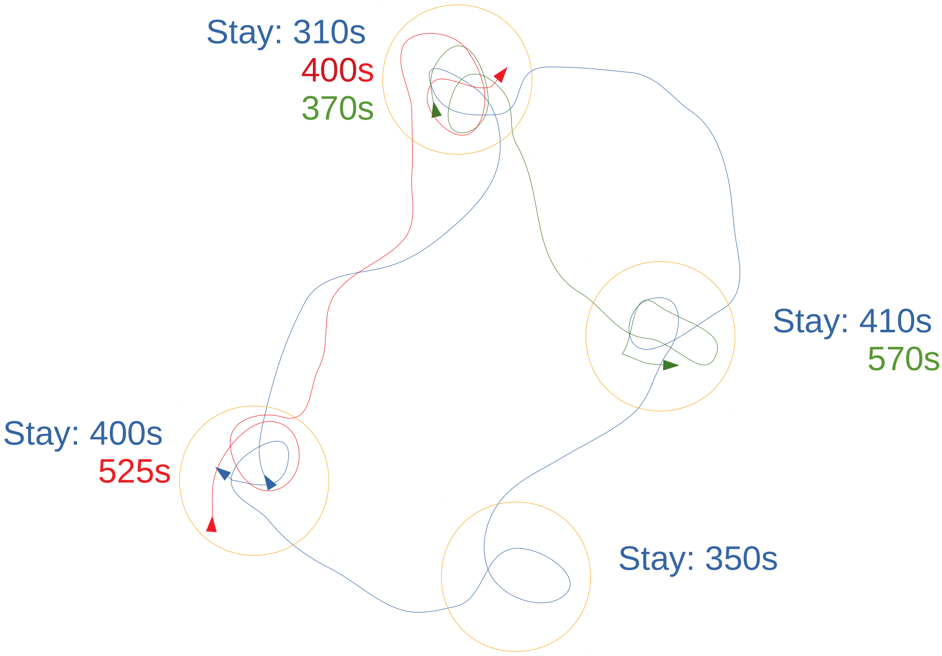
-Then, we check if there is no other user spending a time interval longer than 5 min inside the POI

Algorithm description:



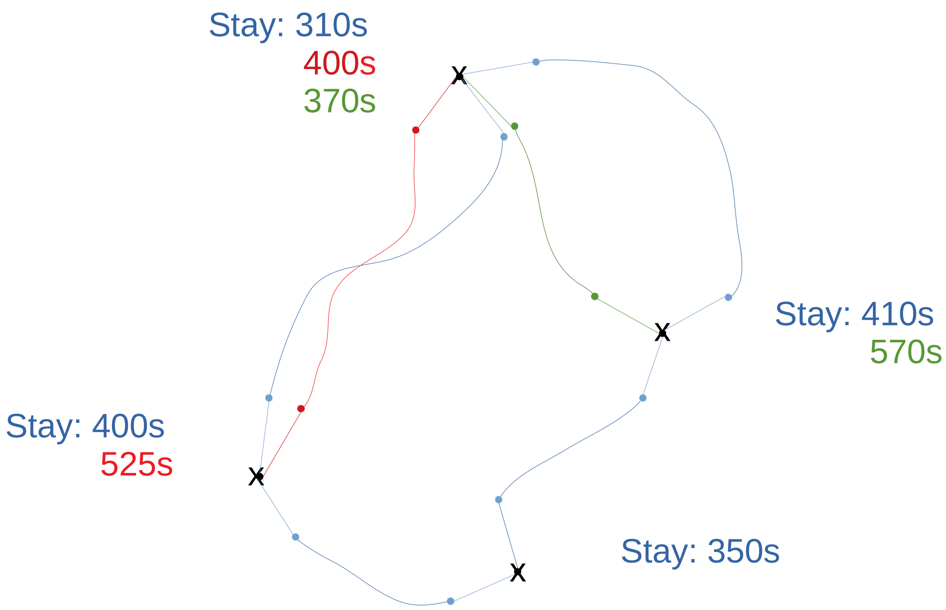
If there is another user spending time inside the POI we add the time interval to a list of the POI

Algorithm description:



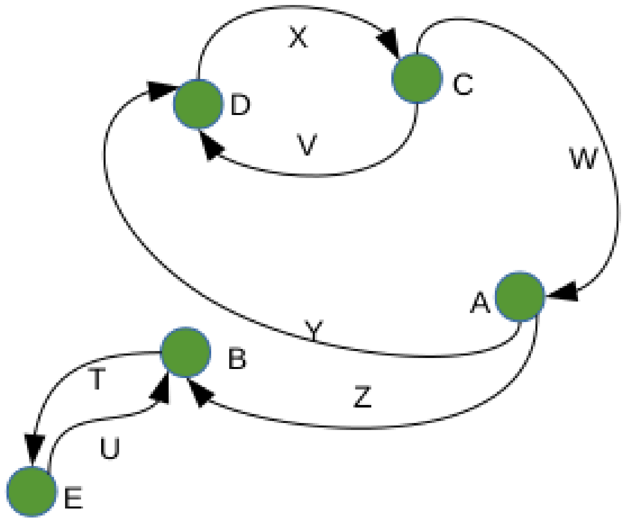
We repeat the process until we find all the POIs in the graph.

Algorithm description:



For each POI we replace all the points inside the enclosing circle by a single point at the center of the circle.

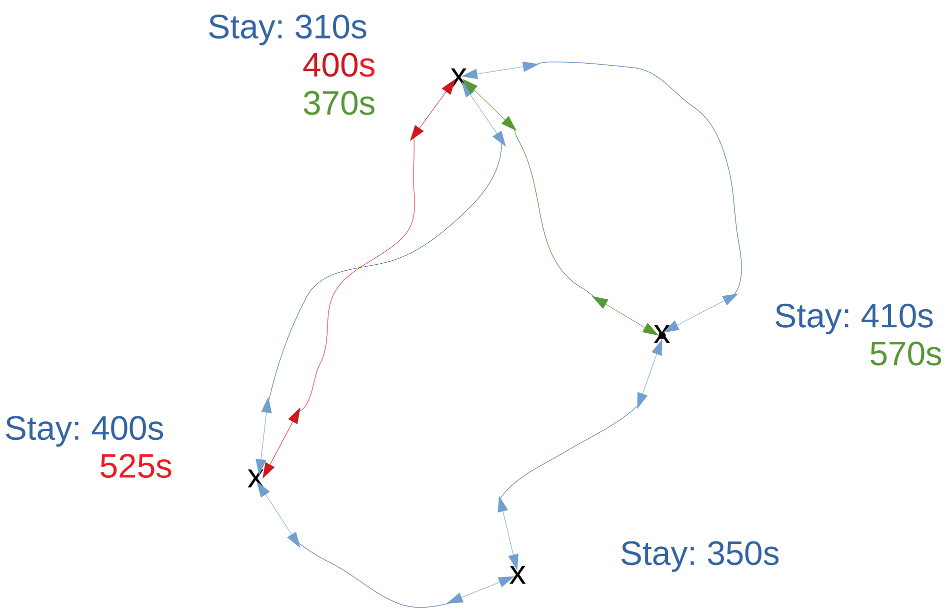
Find links



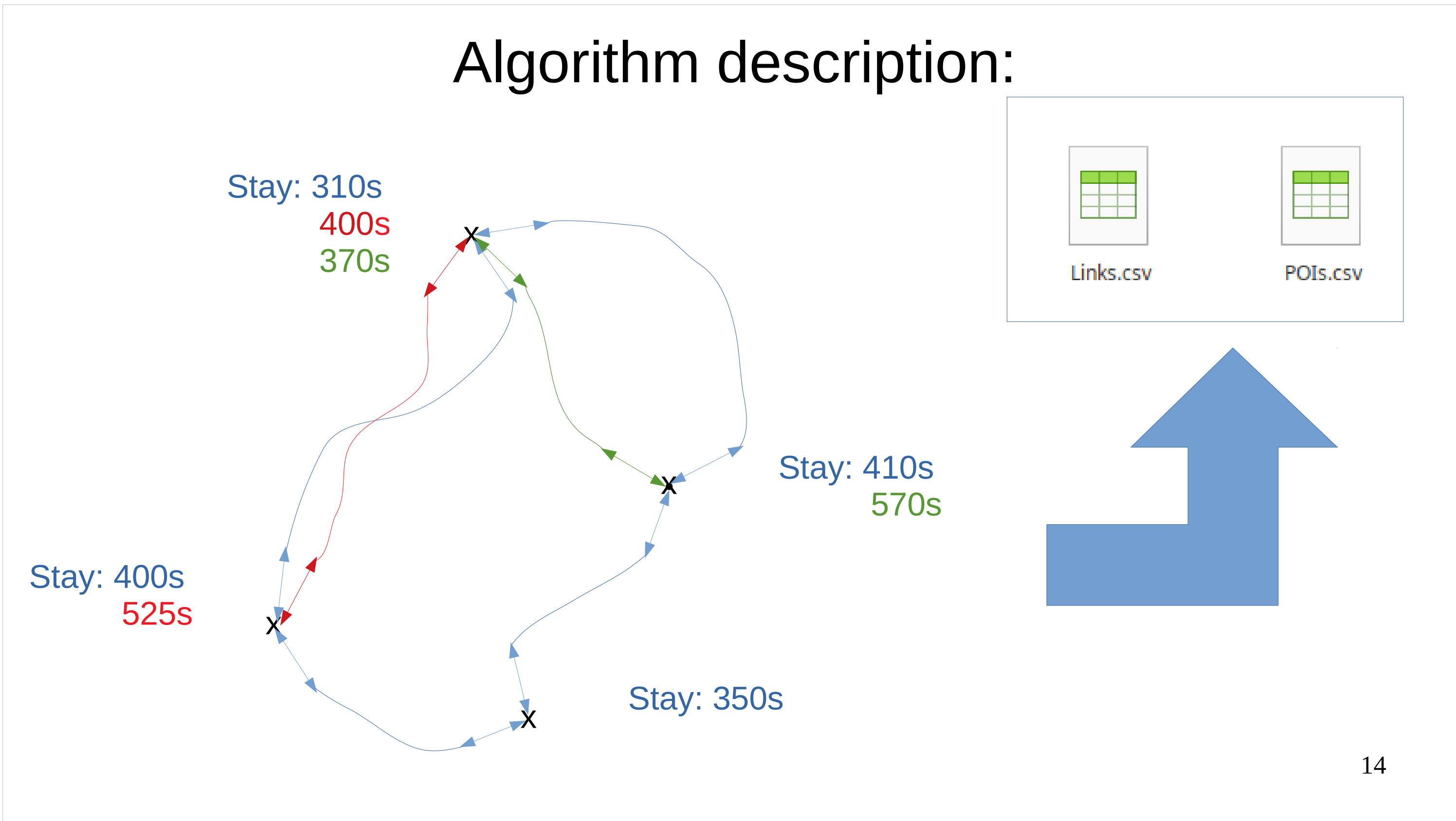
-If we connect the POIs with the Link from the traces as shown in the figure, we would have a problem. For example, if a user travels from A to B it would never go back to A. In other words, if we simulate this graph long enough, all users would end up between B an E.

-To solve this problem we can remove link Z, make link Z bidirectional or make all links bidirectional.

Algorithm description:

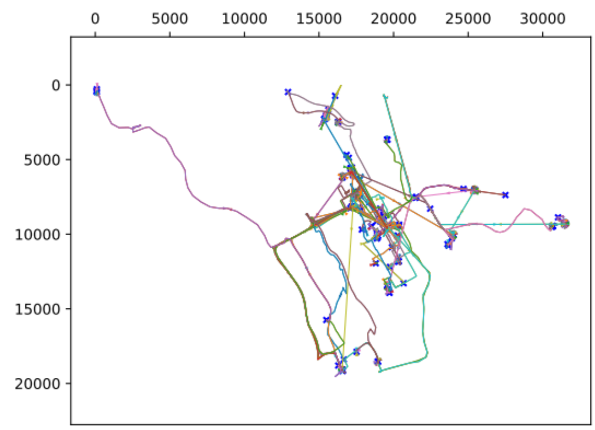


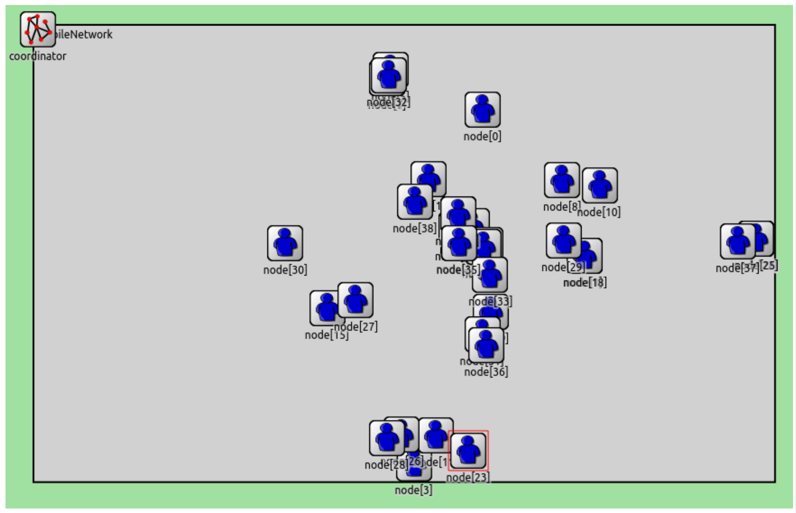
For further analysis we make all links bidirectional Nonetheless, the implemented graph generator is capable of executing any of the 3 solutions.



Finally, we export the POIs and the links to 2 CSV files in order to use them in a simulator as a TRAILS graph.

TRAILS simulator





-The model simulator reproduces users' movements according to a TRAILS model by using a set of rules in combination with a TRAILS graph.

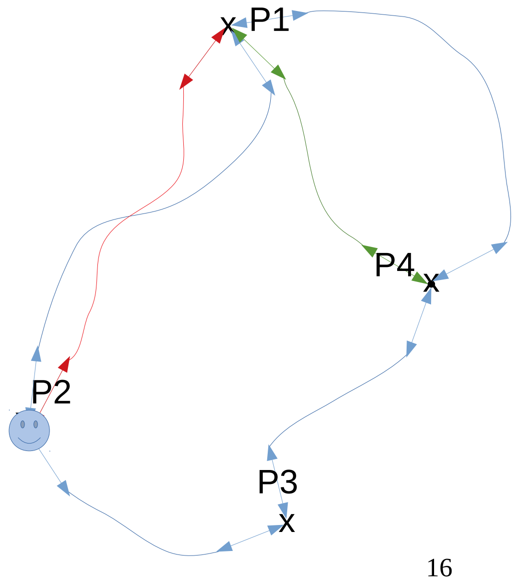
-In the simulator there are mobile users and a coordinator module.

-The coordinator loads the TRAILS graph and it shares the information that each user needs in order to move according to the TRAILS model.

-The next slides describe how the model works in the simulator.

Algorithm description:

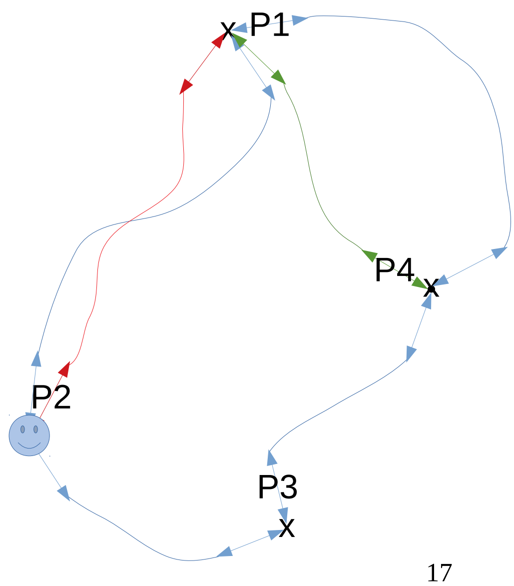
POI	Coordinates	Stay times	Links	Next POI
P1	583;324	400	L1	P4
		310	L2	P4
		370	L3	P2
			L4	P2
POI	**Coordinates**	**Stay times**		
P2	246;820	400	L6	P1
		525	L7	P1
			L8	P3
POI	**Coordinates**	**Stay times**	**Links**	
P3	678;919	350	L9	P4
			L10	P2
POI	**Coordinates**	**Stay times**	**Links**	
P4	743;512	410	L11	P1
		570	L12	P1
			L13	P3



- Once the POIs and links are loaded in the coordinator, the coordinator assigns a random POI to each mobile user.

Algorithm description:

POI	Coordinates	Stay times	Links	Next POI
P1	583;324	400	L1	P4
		310	L2	P4
		370	L3	P2
			L4	P2
POI	**Coordinates**	**Stay times**		
P2	246;820	400	L6	P1
		525	L7	P1
			L8	P3
POI	**Coordinates**	**Stay times**	**Links**	
P3	678;919	350	L9	P4
			L10	P2
POI	**Coordinates**	**Stay times**	**Links**	
P4	743;512	410	L11	P1
		570	L12	P1
			L13	P3



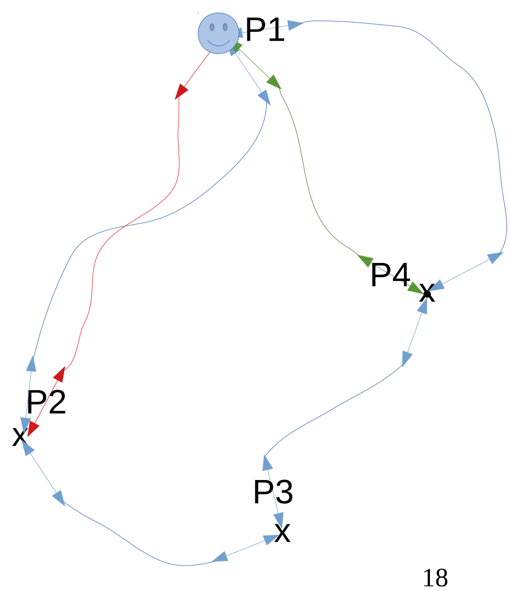
-When all users are in their initial position. The coordinator sends a link and a stay time, chosen randomly.

-Then, the user waits in its position for a time interval equal to the stay time.

-Finally, the user follows the link until it arrives to its next POI.

Algorithm description:

POI	Coordinates	Stay times	Links	Next POI
P1	583;324	400	L1	P4
		310	L2	P4
		370	L3	P2
			L4	P2
POI	**Coordinates**	**Stay times**		
P2	246;820	400	L6	P1
		525	L7	P1
			L8	P3
POI	**Coordinates**	**Stay times**	**Links**	
P3	678;919	350	L9	P4
			L10	P2
POI	**Coordinates**	**Stay times**	**Links**	
P4	743;512	410	L11	P1
		570	L12	P1
			L13	P3



- When a user arrives to its next POI, it sends the index of the POI to the coordinator.
- The coordinator sends back a new stay time and link randomly chosen, and the cycle is repeated.

-It can be observed that by this model, TRAILS simulations are not limited by a fixed number of users or a fixed simulation time.

Results & Conclusions

19

-We simulated TRAILS and traces and then we used statistical tests to observe the relation of TRAILS with real scenarios.

TRAILS vs Traces

Metrics

1. Contact duration between nodes (**CDBN**)
2. Contact probability (**CNPR**)
3. Number of of contacts between nodes (**NCBN**)
4. Number of of contacts between the same pair of nodes (**NCBS**)
5. Contact time between nodes (**CTBN**) (Intercontact)

Scenarios

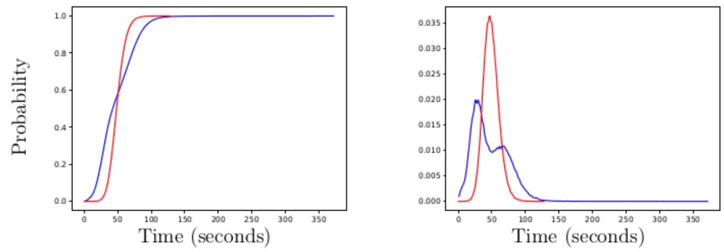
Name	Area Km^2	Time Hours	Users
San Francisco	7145.486	575.425	536
Rome	10238.67	720	315
New York	618.699	22.658	39
Orlando	276.592	14.283	41

The performance of OppNets depends on the interaction between users. Therefore, we use the following metrics to compare sample distributions of TRAILS and traces.

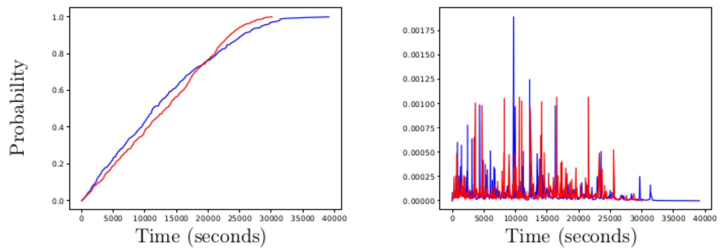
CDBN is the time interval when two users are at a distance smaller than 30m.
CNPR is the estimation of the probability of two nodes in contact.
NCBN is the total number of times that each user was in contact with another user.
NCBS is the total number of times that a pair of users were in contact.
CTBN is the time interval between the last contact and a new user contact (intercontact).

In order to avoid biased conclusions we used 4 different scenarios

San Francisco



NCBS of TRAILS (red) and traces (blue) of 536 taxis in San Francisco



NCBN of TRAILS (red) and traces (blue) of 536 taxis in San Francisco

21

- -By only looking at the sample distributions it is difficult to know how different TRAILS is from traces. The only conclusion that we can reach from the plots is that the samples are not normally distributed.
- -Therefore, we need to perform a statistical test to the metrics to have a quantitative result.

San Francisco

Mann–Whitney U test

Metric	ρ-score	p-value
CDBN	11.106	$1.1715e^{-128}$
CNPR	124.379	0.0
NCBS	54.732	0.0
NCBN	1.273	0.202
CTBN	45.575	0.0

Our Null hypothesis establishes that both samples are totally different or unrelated.

Probability value (p-value) should be smaller than 0.05 (5%) to accept the hypothesis.

-The MW-U test is a nonparametric statistical test which evaluates how likely is that two random variables belong to the same population.
-The MW-U test is equivalent to the T test for normal distributions.
- If we observe the results of the test, only one parameter is not totally unrelated.

- I chose a standard nonparametric test instead of comparing confidence intervals because comparing intervals is less accurate than hypothesis tests, according to Mark Payton in his paper "*Overlapping confidence intervals or standard error intervals: What do they mean in terms of statistical significance?*".
-Another reason is that the p-value of a test shows how probable is that 2 random variables belong to the same population.

Statistical Test

Rome

Metric	ρ-score	p-value
CDBN	-68.646	$1.1715e^{-128}$
CNPR	57.471	0.0
NCBS	49.73	0.0
NCBN	-0.0752	0.94
CTBN	40.737	0.0

San Francisco

Metric	ρ-score	p-value
CDBN	11.106	$1.1715e^{-128}$
CNPR	124.379	0.0
NCBS	54.732	0.0
NCBN	1.273	0.202
CTBN	45.575	0.0

New York

Metric	ρ-score	p-value
CDBN	2.329	0.0198
CNPR	-0.182	0.854
NCBS	-0.259	0.795
NCBN	2.145	0.0319
CTBN	-4.61	$4.0116e^{-06}$

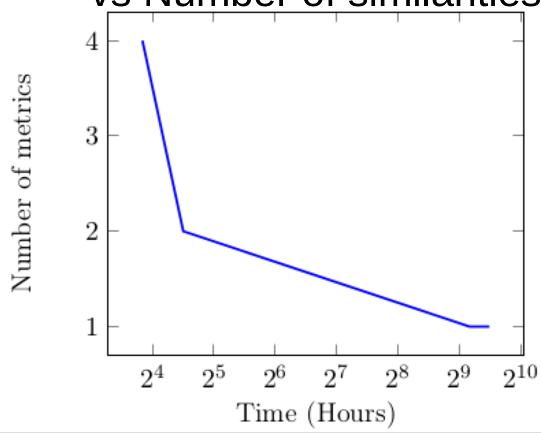
Orlando

Metric	ρ-score	p-value
CDBN	0.462	0.643
CNPR	0.87	0.383
NCBS	0.906	0.364
NCBN	1.587	0.112
CTBN	2.444	0.0145

Scenarios

Name	Area Km^2	Time Hours	Users
San Francisco	7145.486	575.425	536
Rome	10238.67	720	315
New York	618.699	22.658	39
Orlando	276.592	14.283	41

Relation Record time vs Number of similarities



-If we look at the results of the tests of the 4 scenarios, we can see that scenarios with a short recording time can be represented better by TRAILS than scenarios with a long recording time.

-The reason is that mobile users change their behavior over time. For example, people have different routines in the day than in the night.

-Other mobility models like CMM address this problem by changing the model periodically, for example every 8 hours. On the other hand, we are using the same TRAILS graph to represent a very long trace.

-In chapter 6 of my thesis in the section called Time invariability test we prove how metrics on traces change over time and how these changes affect our validation results.

Performance

Scenario	Processed traces	TRAILS graphs
San Francisco	474.2	357.5
Rome	755.3	559.8
New York	0.431	0.202
Orlando	1.3	0.576

Size in Mega Bytes of mobility graphs

< 43.37%

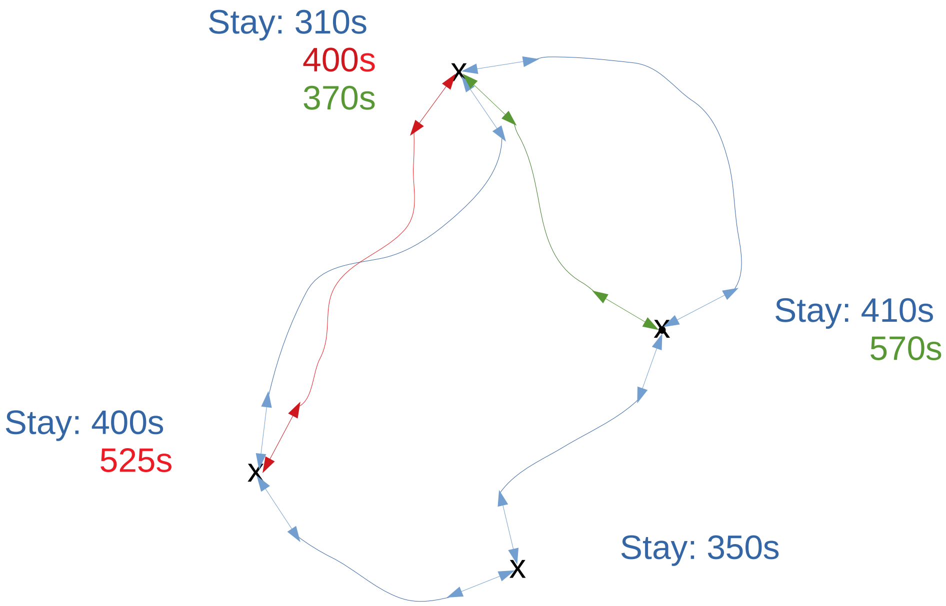
Scenario	Traces	TRAILS
San Francisco	82273.985	63318.721
Rome	35337.649	24466.268
New York	83.208	60.397
Orlando	56.683	40.563

Time spent in seconds by simulating mobility graphs

< 28.1%

- -TRAILS graphs are in average 43.37% smaller than traces.
- -TRAILS simulation requires in average 28.1% less time than traces.
- -The reason is that TRAILS replaces several trace-points by a single POI and this action reduces the size of the graphs and the simulation time.

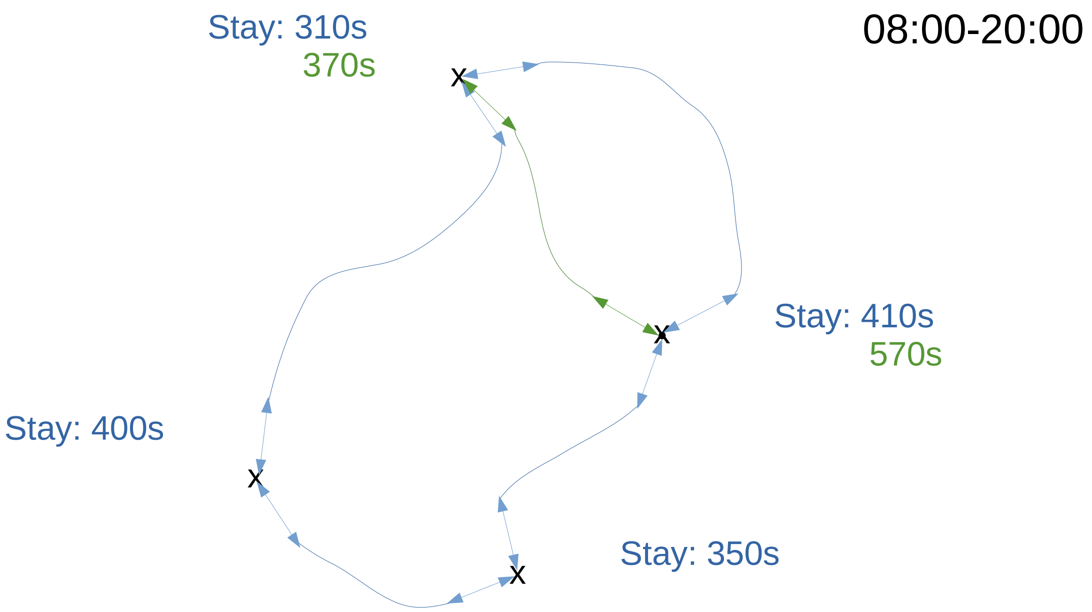
Recommendations



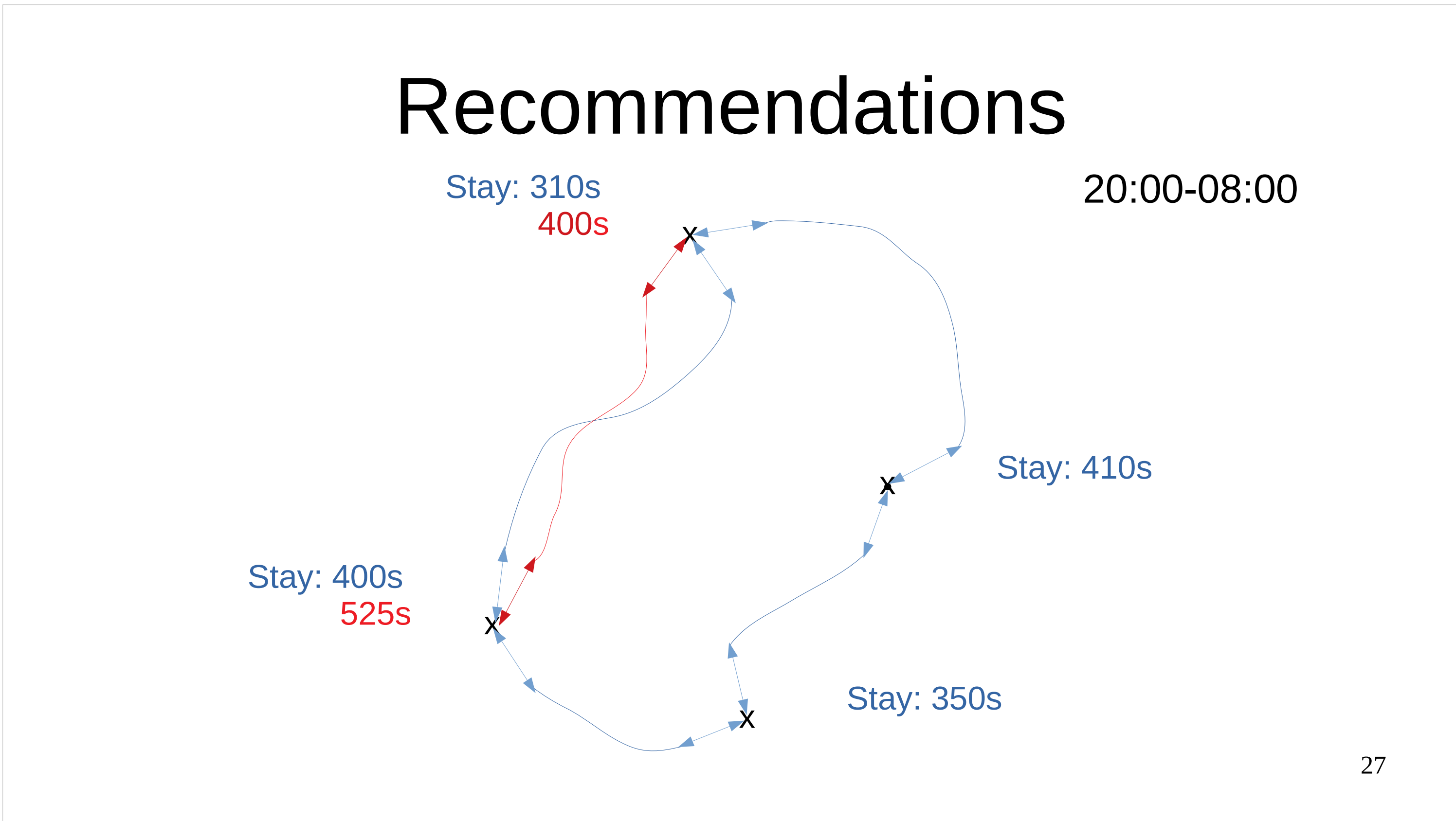
-To increase the similarities of TRAILS with long traces we propose to have different links and POIs available at different periods of time.

-For example if this was the complete TRAILS graph...

Recommendations



-… this would be the graph from 8 am until 8 pm...



-… and this would be the graph from 8pm to 8am.

-By using this strategy we would be able to reduce the differences of the users' interactions between TRAILS and real scenarios.

Thank you!