

An aerial, black and white photograph of a city street intersection. A white line highlights a path that starts on the left side of the road, turns right at a crosswalk, and then continues straight. The path is highlighted in white against the dark asphalt. The background shows a multi-lane road with lane markings, a crosswalk, and a building with a sign. The overall scene is captured from a high angle, looking down at the street.

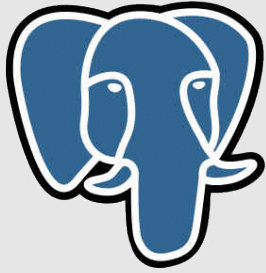
# Shortest Path Search with *pgRouting*

Daniel Kastl  
*Georepublic*



What is pgRouting?

PostgreSQL



An **Extension** for  
PostgreSQL / PostGIS, ...



An Open Source  
project, ...



A **Library** providing, ...



Shortest Path



DD and TSP



DARP





@justjkk

@jay\_mahadeokar



APSP and TDSP



Multi-Modal

Fork me on GitHub



<https://github.com/pgRouting/pgrouting>

# SQL Function

```
SELECT * FROM shortest_path(  
    SELECT gid as id,  
           source::integer,  
           target::integer,  
           length::float8 as cost  
    FROM ways',  
    605, 359, false, false);
```

# Query Result

vertex_id	edge_id	cost
605	599	0.19925085940845
604	598	0.100258103875674
603	597	0.201123583387407
602	596	0.204847680679676
601	595	0.158334540345002
4293	9602	0.0183273901669081
...	...	...

Most users need pgRouting for  
**Road Networks**

How do they look like?



Like this ...







... or like this ...



... or sometimes like this.



# What makes them real?



Traffic lights



Signs



Road marking

How can **pgRouting** help here?





source

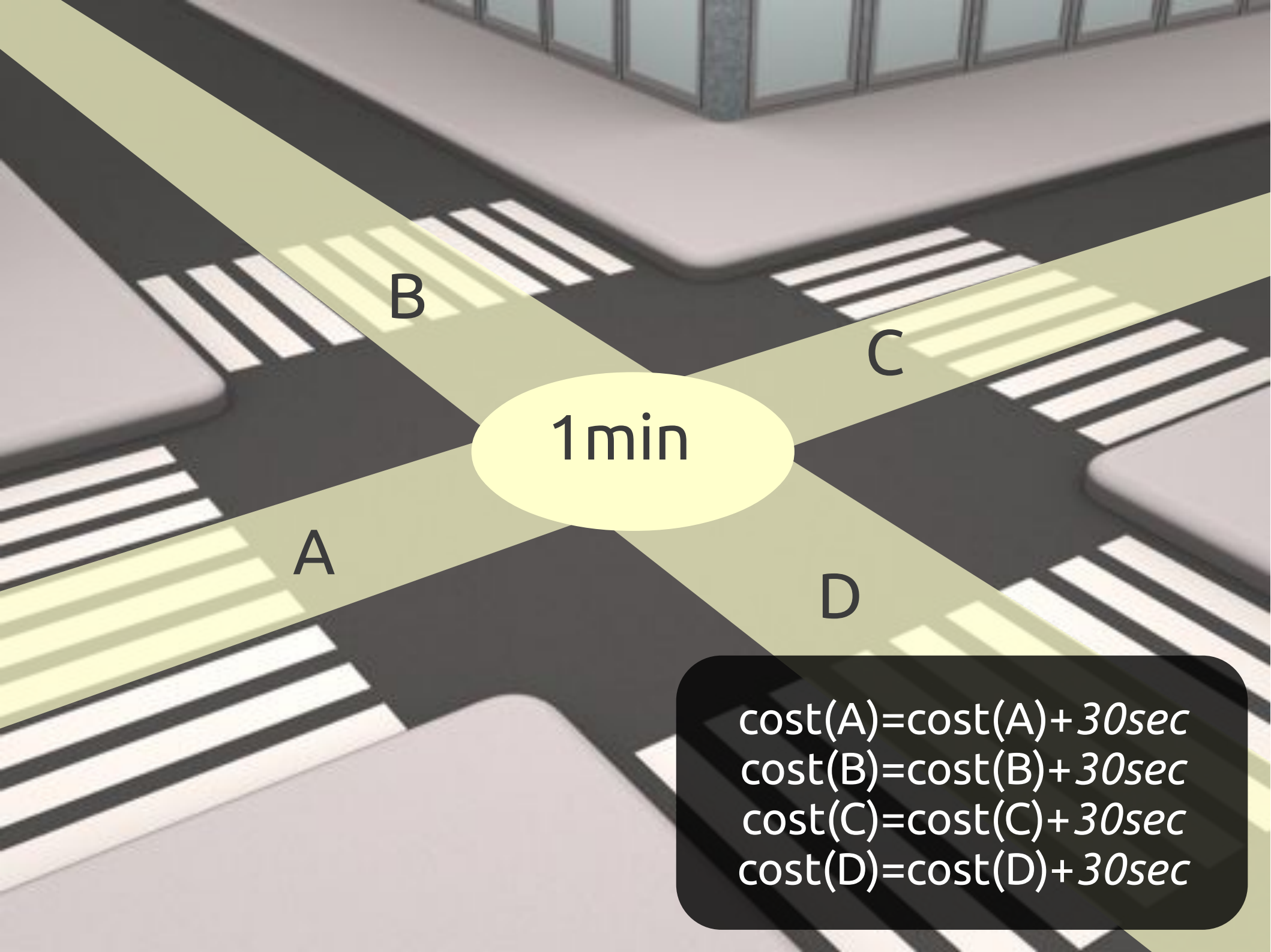
target

gid,  
source,  
target,  
cost,  
reverse\_cost,  
x1, y1,  
x2, y2,  
rule,  
to\_cost

# Traffic lights slow down



... so costs must increase.



B

C

1min

A

D

$\text{cost}(A) = \text{cost}(A) + 30\text{sec}$   
 $\text{cost}(B) = \text{cost}(B) + 30\text{sec}$   
 $\text{cost}(C) = \text{cost}(C) + 30\text{sec}$   
 $\text{cost}(D) = \text{cost}(D) + 30\text{sec}$



Signs inform about  
restrictions and rules







**ONE WAY**

$\text{cost}(A) = \text{length}(A)$   
 $\text{reverse\_cost}(A) = \infty$

A photograph of a city street with a yellow arrow pointing down the road, labeled 'A'. The street is lined with multi-story buildings on both sides. A yellow arrow starts from the bottom left and points towards the center of the street, with the letter 'A' written below it. A horizontal yellow line with a diamond-shaped arrowhead at its right end points from the center of the street towards the right side of the image.

A

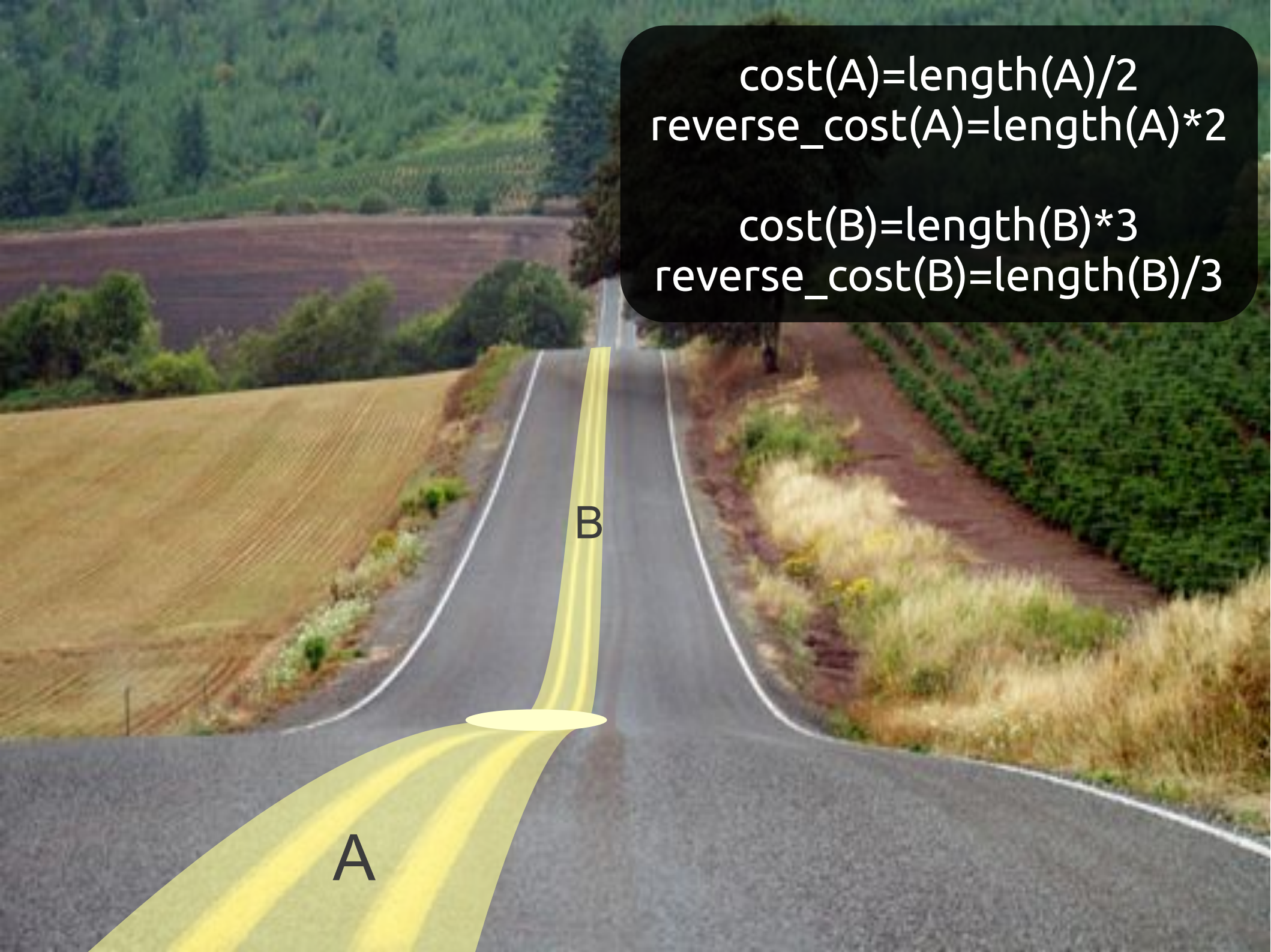


Sometimes  
the costs  
have *different*  
meaning.



$\text{cost}(A) = \text{length}(A) / 2$   
 $\text{reverse\_cost}(A) = \text{length}(A) * 2$

$\text{cost}(B) = \text{length}(B) * 3$   
 $\text{reverse\_cost}(B) = \text{length}(B) / 3$



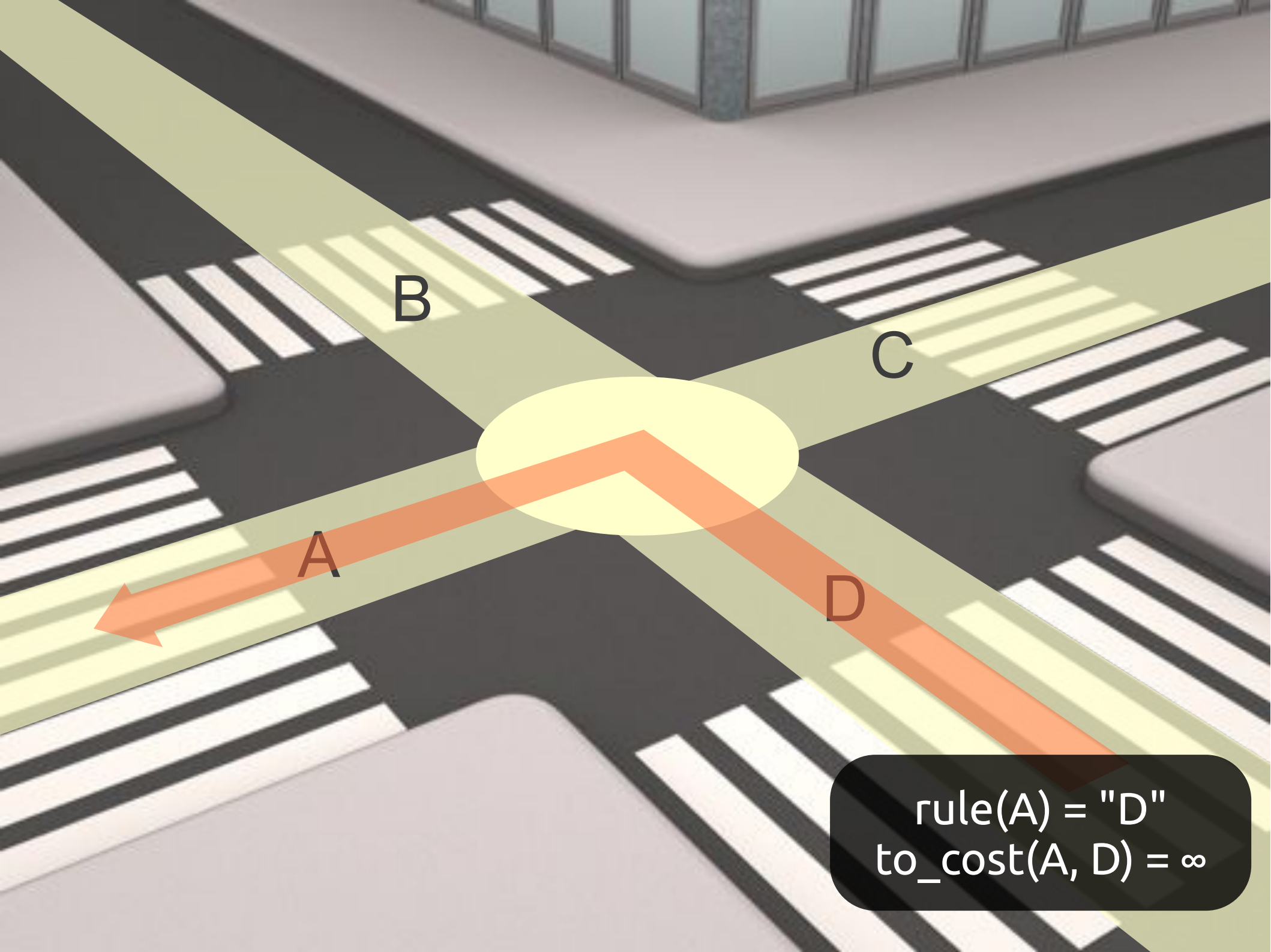
B

A

*Turn restrictions*  
obviously  
restrict turns.







B

C

A

D

rule(A) = "D"  
to\_cost(A, D) =  $\infty$

*Road type*  
can be used  
for cost  
calculation.





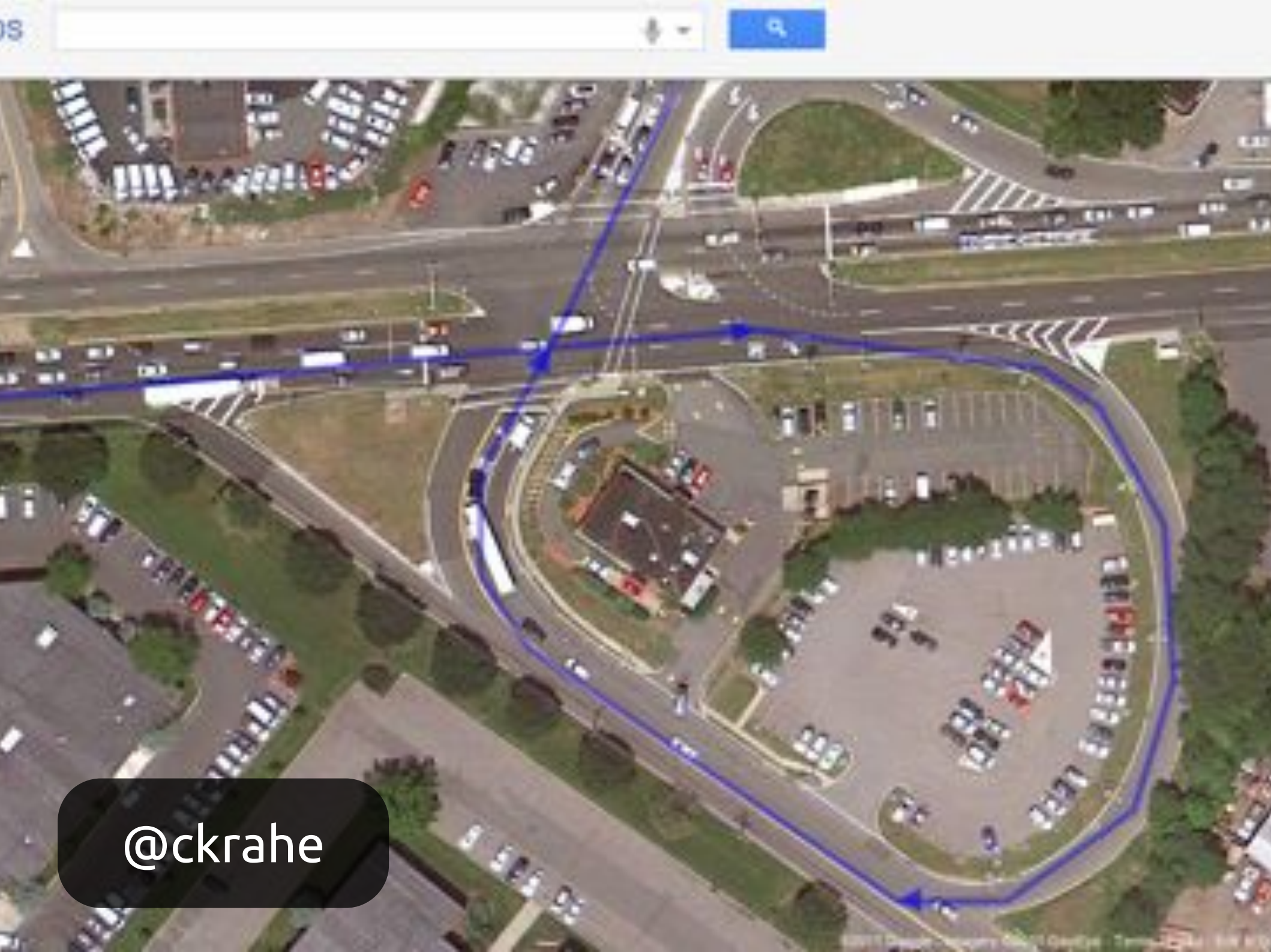
Not only road *types*  
but also *conditions*





# Jughandle intersection





@ckrahe

And you know  
what is *great* about pgRouting?

All costs are *dynamic*!

... which is the opposite to pre-calculated



If the road is closed ...

**ROAD  
AHEAD  
CLOSED**





... or there is an accident, ...



... there is a sign with restrictions limited to a certain time, ...

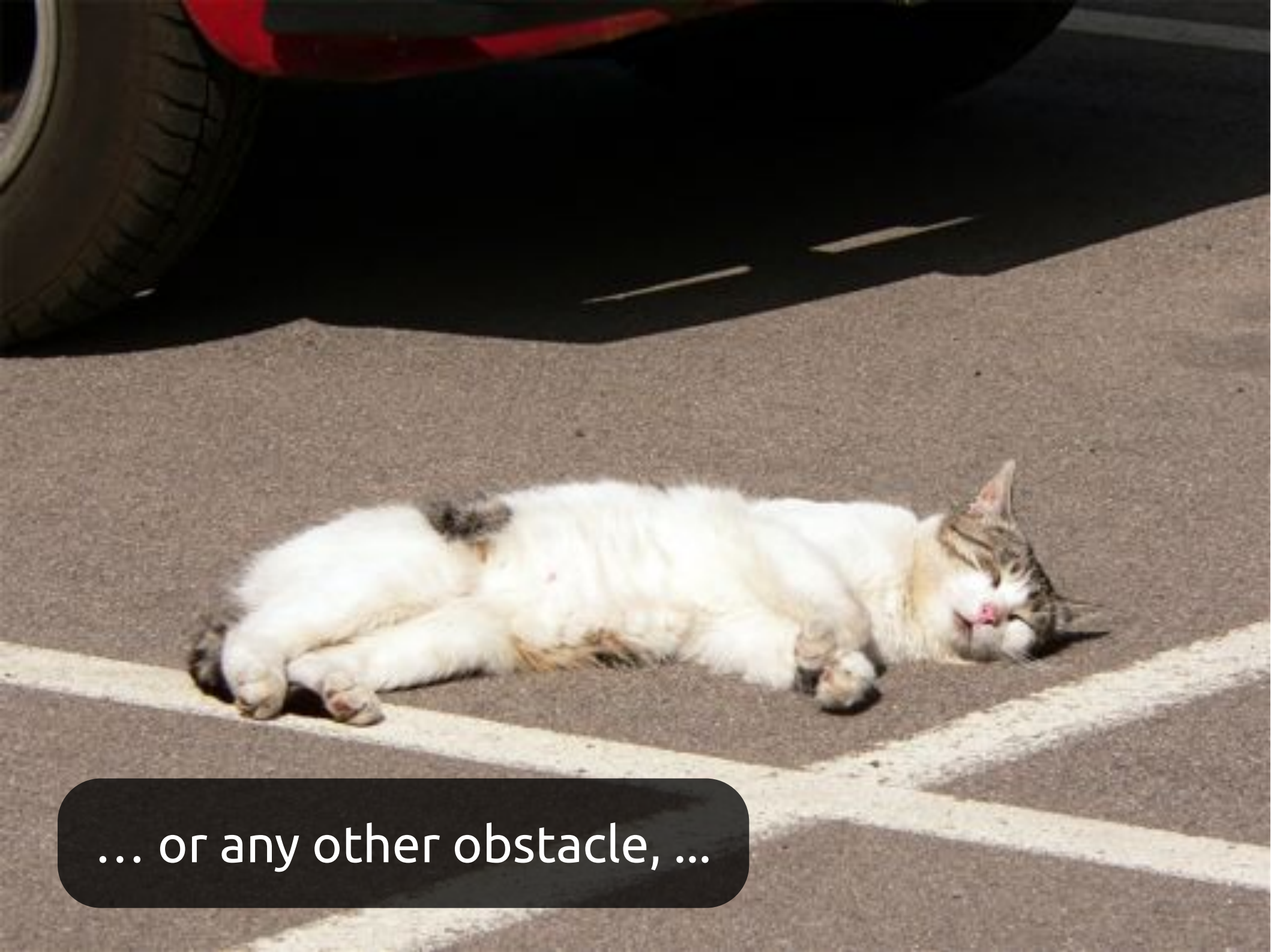






... bad weather conditions ...





... or any other obstacle, ...



You don't need to  
rebuild and reload  
your network





... and wait forever.

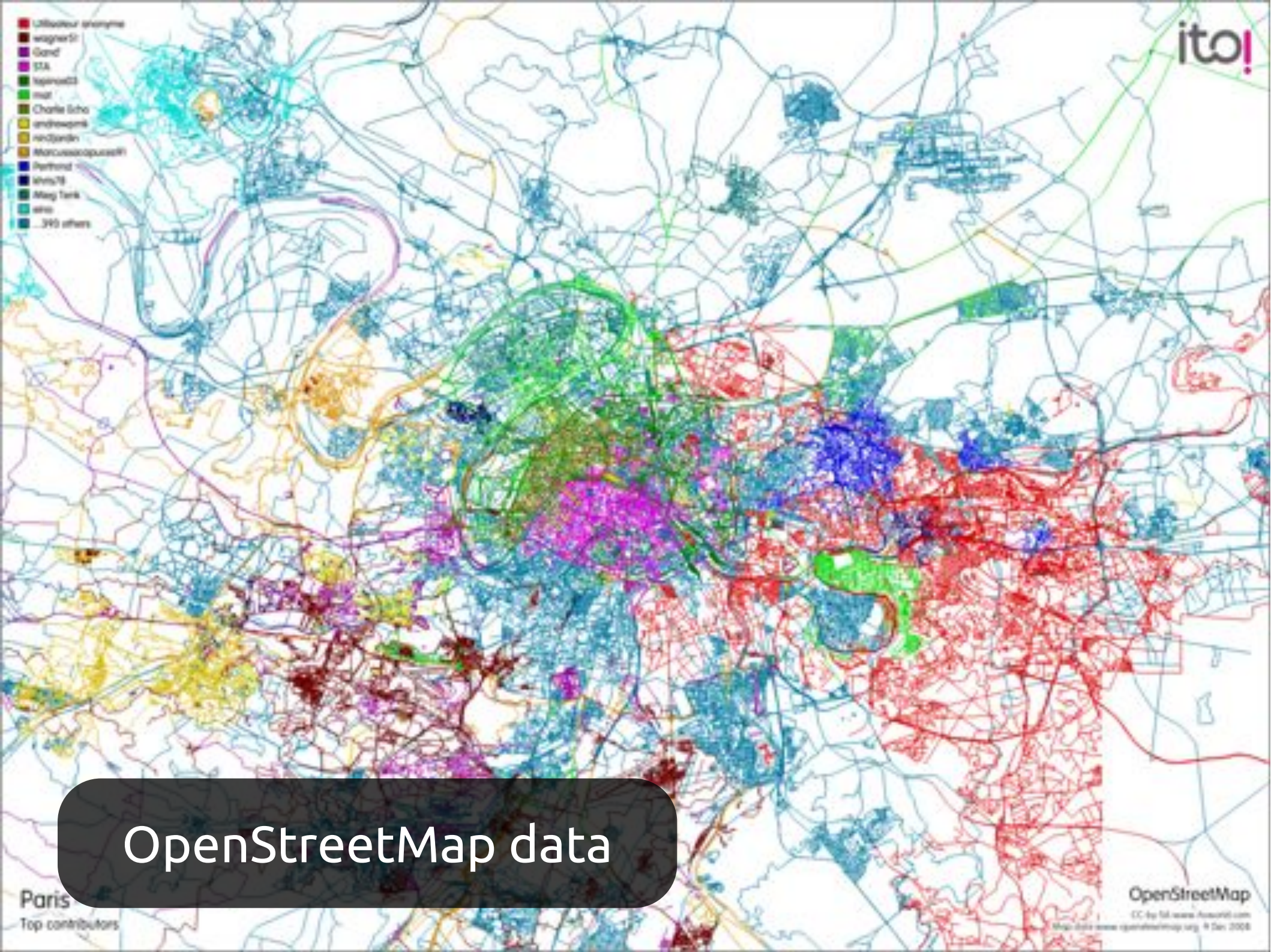
You only need to adjust the cost  
for this particular road,  
and the next search will go another way.



Cost can be virtually anything



- Ullricher anonyme
- wagner51
- Gand
- STA
- lapin023
- nut
- Charlie Echo
- andrewpb
- incjardin
- MarcusScopuseff
- Perthois
- shu78
- Mog Tank
- shu
- ...390 others



# OpenStreetMap data

pgRouting can be used for  
*different* kinds of networks



# Canals and Rivers



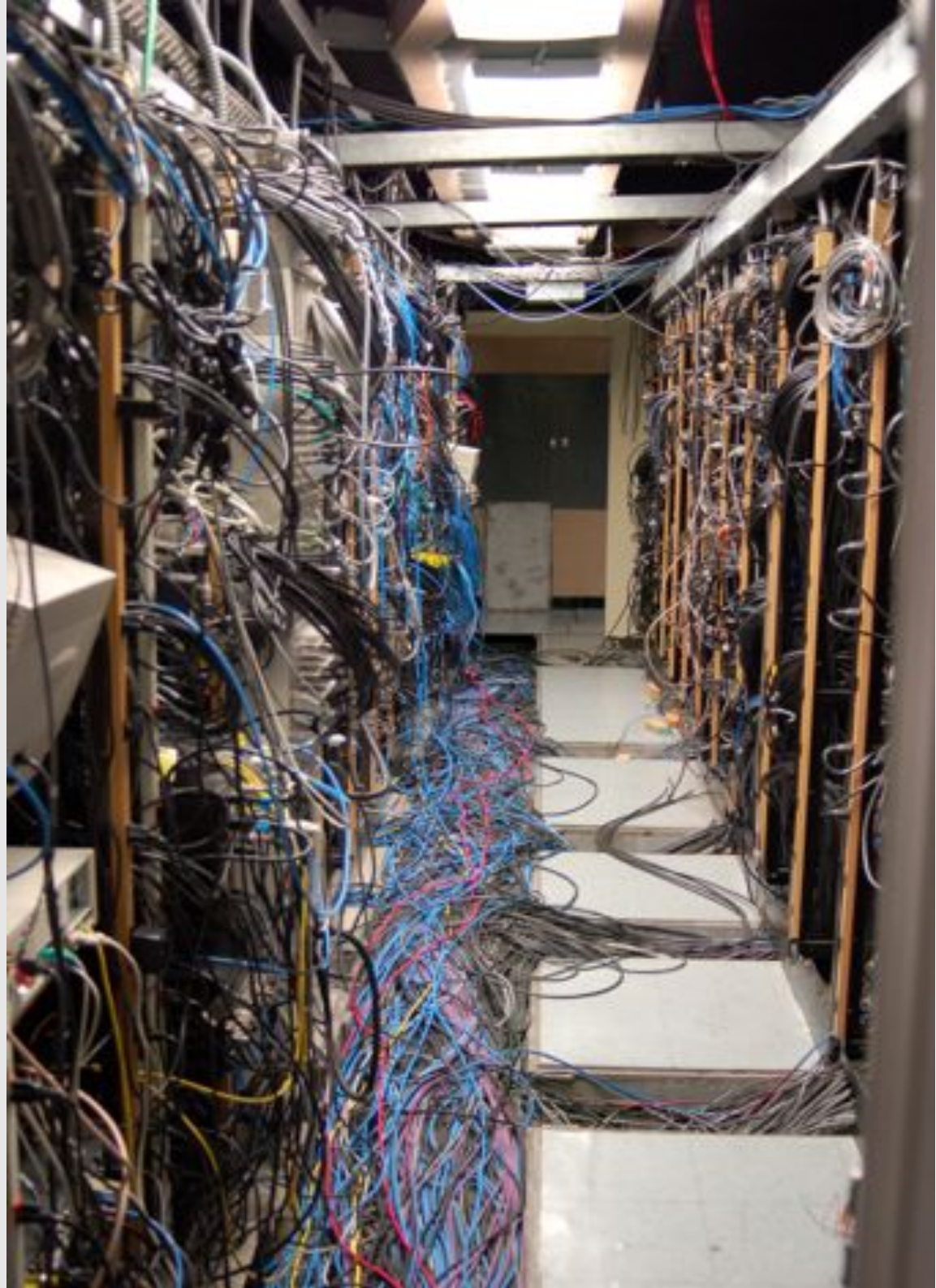


# Hiking trails





... or any other  
kind of networks.



# pgRouting Demo

<http://websi.openvrp.com>

<http://map.veloland.ch>

<http://www.ridethecity.com>

<http://www.pgrouting.org/gallery.html>

# Plans until FOSS4G 2012

- Integrate new functions
  - All-Pair-Shortest-Path
  - Time-Dependent-Shortest-Path
  - Multi-Modal
  - Two-way A-Star, ...
- Drop CGAL dependency
- PostgreSQL 9.x & PostGIS 2.0 support
- Source cleanup





# pgRouting

[Home](#) | [Documentation](#) | [Download](#) | [Support](#) | [Development](#)

Search

## pgRouting Project

pgRouting extends the [PostGIS](#) / [PostgreSQL](#) geospatial database to provide geospatial routing functionality.

Advantages of the database routing approach are:

- Data and attributes can be modified by many clients, like [Quantum GIS](#) and [uDig](#) through JDBC, ODBC, or directly using PL/pgSQL. The clients can either be PCs or mobile devices.
- Data changes can be reflected instantaneously through the routing engine. There is no need for precalculation.
- The "cost" parameter can be dynamically calculated through SQL and its value can come from multiple fields or tables.



## Download

Current release: [1.05](#)

## Make a Donation

[Donate](#) EUR ▾

Support pgRouting with some donation! [Read more](#) about how your donation will help the project.

## Links

- [pgRouting Workshop](#)
- [PostgreSQL](#)
- [PostGIS](#)



## Core Features

pgRouting

- Shortest Path Dijkstra: routing algorithm without heuristics
- Shortest Path A-Star: routing for large datasets (with heuristics)
- Shortest Path A-Star: routing for large datasets (with heuristics)
- Traveling Salesperson Problem (TSP)
- Driving Distance calculation (Isolines)

[www.pgrouting.org](http://www.pgrouting.org)

[workshop.pgrouting.org](http://workshop.pgrouting.org)

[daniel@georepublic.de](mailto:daniel@georepublic.de)



A photograph of a road divider with a sign that says "THANK YOU". The divider is covered in fallen leaves. The road curves to the right in the background, lined with trees. The sign is white with black text and is mounted on a metal post. The background shows a dense line of green trees and a clear sky.

THANK YOU

Photos from *sxc.hu* and  
**flickr** under *Creative Commons* Licence.