TSUT:
The **S**till **U**nnamed **T**ool for mesh network planning

Leonardo Maccari
leonardo.maccari@unitn.it
leonardo.maccari@unive.it

whois Leonardo

- I am a professor at the University of Venice
- Till May I was a researcher at the university of Trento (Italy)
- I am a member of the ninux.org network in Florence
- I was the WP technical coordinator of the netCommons project, a three-year H2020 research project on CNs that ended in March 2019

The netCommons Project: 2016-2019

UNIVERSITY OF TRENTO - Italy

UPC

cnrs dépasser les frontières

NetHood

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

UNIVERSITY OF WESTMINSTER

- H2020 Financed project (CAPS)
- 2016-2019
- 4 Universities
- 1 Research Center
- 1 not-for-profit association
- 6 countries

netCommons: what we did

- Under a global point of view:
  - we influenced the EU legislation mechanism to be more CN-friendly
  - we convinced UNESCO to include CNs in the way they evaluate national ICT policies
- Under a local point of view:
  - We described how several CNs work, their sustainability and governance
  - We contributed to the development of some
  - We analysed the technical evolution of some
  - We also contributed with open source code, guides etc.
  - ...

`www.netcommons.eu`

TSUT: The Still Unnamed Tool

- TSUT was not initially part of the project, it came out as an idea in the process
- It has a double nature:
  - Research: generate and study realistic network topologies of a mesh network.
  - Communities: help to plan your network
- Three components:
  1. Open data surface models
  2. Radio models taken from data-sheets and some literature
  3. An engine that simulates the growth of the network

Warning

Current state:

- Python code on github[a], but really to be revised (realized in a rush for a deadline. . .)
- Quite complex, there are a lot of different components (postgres/postgis, networkx), partial test coverage
- A lot of heuristics in our model, which we will hopefully improve in the future
- Consider this as a Proof of Work

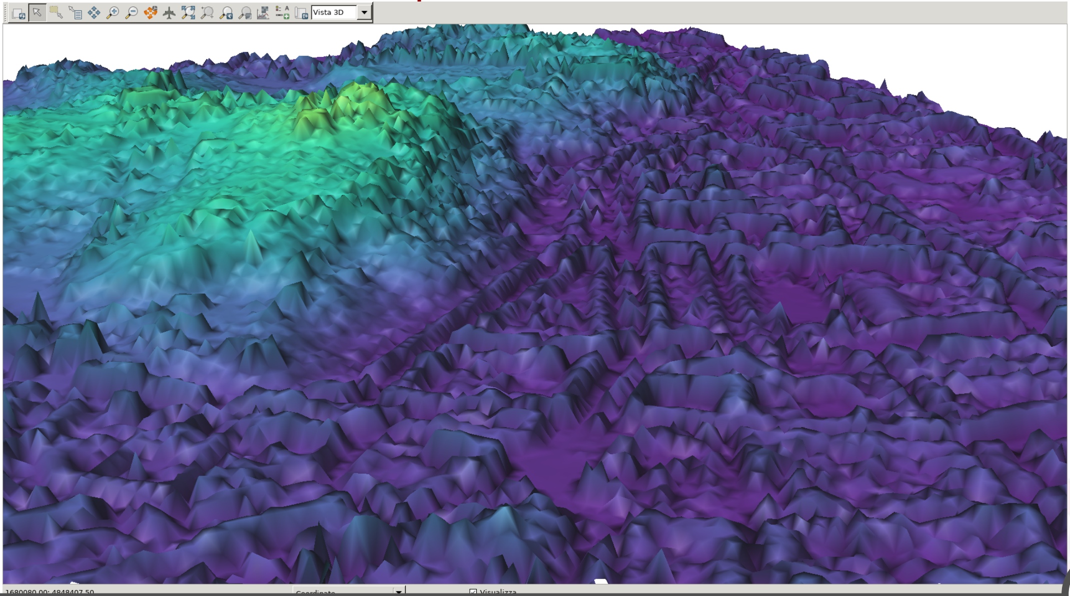
[a] `https://github.com/AdvancedNetworkingSystems/TerrainAnalysis`

Dataset

- We start from the open data-set of the building altitudes of an area (Lidar data)
- We add the building shapes taken from OpenStreetmap/Catasto
- For each couple of buildings, we can compute:
  - If there is Line of Sight
  - If the Fresnel zone is partially obstructed
  - How high is the path loss considering the Fresnel occupation

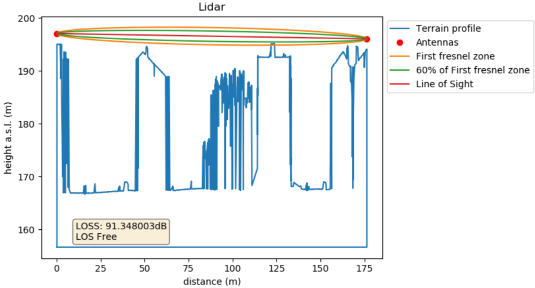
A CN simulator: Lidar data

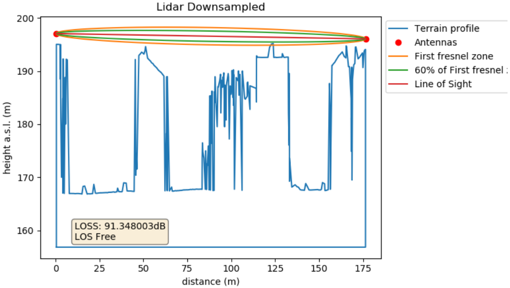
A CN simulator: altitude profiles

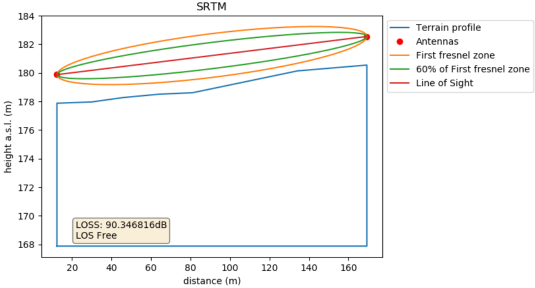


A CN simulator: Lidar + OS

A CN simulator: Fresnel zone with Different Sampling



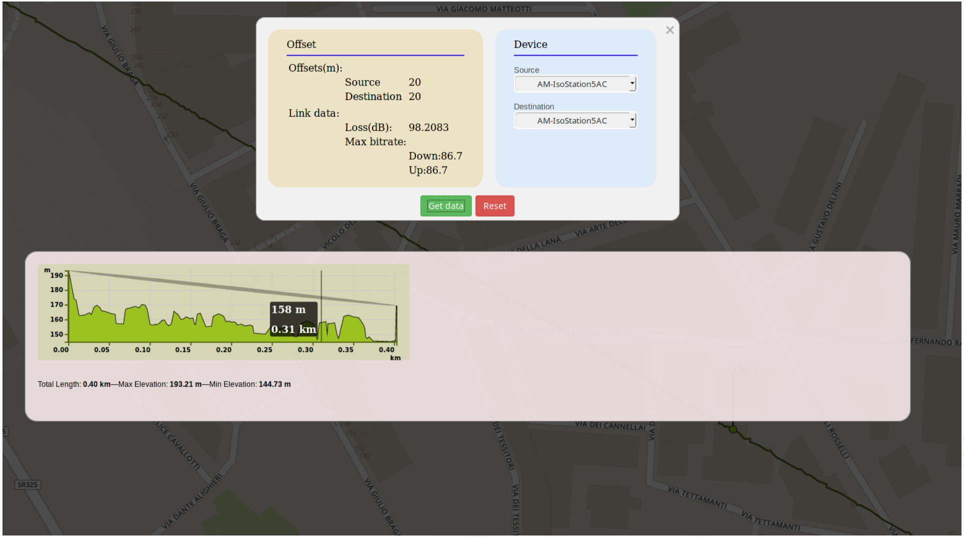




A Database of Devices

- We collected the data-sheets of Ubiquiti devices (July 2018)
- Given the path loss, we can choose the most appropriate device according to some criteria (highest bit-rate, lowest cost, narrow antenna aperture. . .)
- We assume Point-to-point links, and can estimate the cost of each link/node

A very simple web interface



A very simple web interface

click me in case you don't remember the URL

Growth Heuristics

- We decide the location of a network gateway, and we pick a sequence of random buildings in the area
- We connect each now node to some existing one
- We need to model the maximum available bandwidth per node in saturation conditions: the "guaranteed bandwidth per user"
- This involves a number of heuristics to model the routing decision, channel allocation, bandwidth/txpower negotiation...

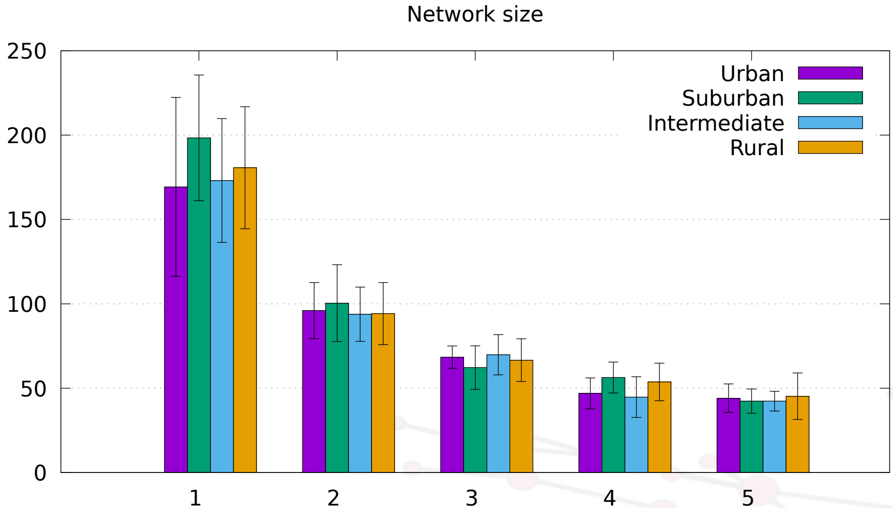
Stop Condition

- Once we can estimate the minimum bandwidth to the gateway per node, we need a stop condition
- The stop condition is: stop growing when $x\%$ of the nodes have less than B_{min} Mb/s guaranteed

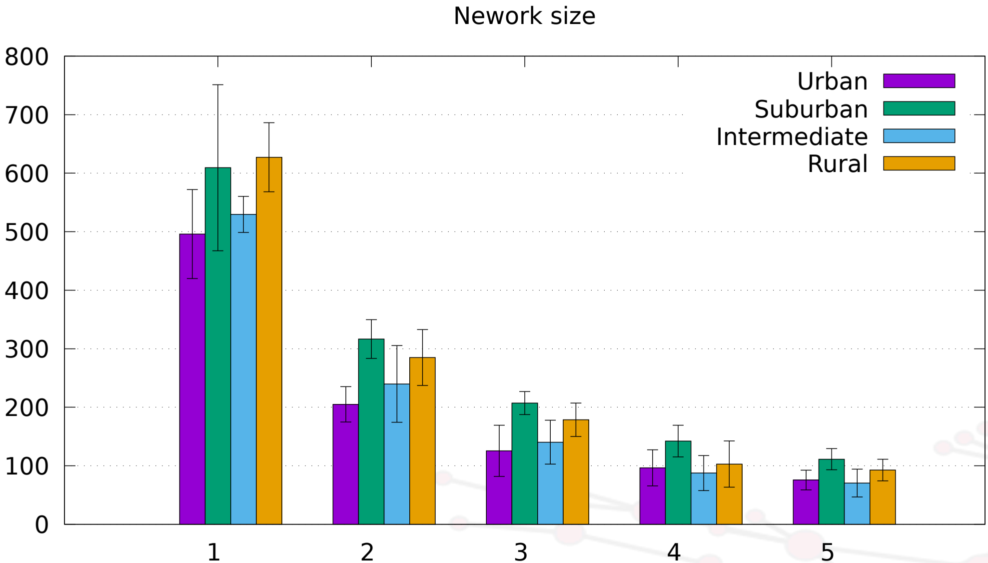
What research we do with TSUT

- Simulate how much such a network can scale
- Given a new node, suggest a reasonable attachment algorithm: what is the best neighbour to connect a new node?
  - Greedy: The one that gives you the best link bandwidth
  - Network-aware: The one that better distributes the load on the gateway
- Examples: map, animation.

Growth of one network: Averaege Size (10 runs), Greedy approach



Network Size: network-aware attachment

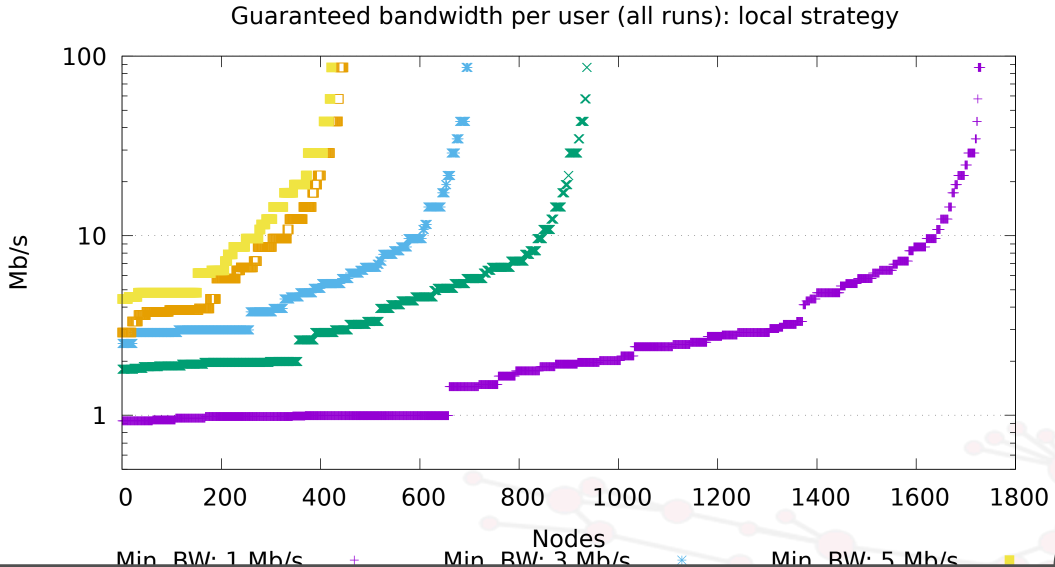


- Thank you for your attention
- Questions?

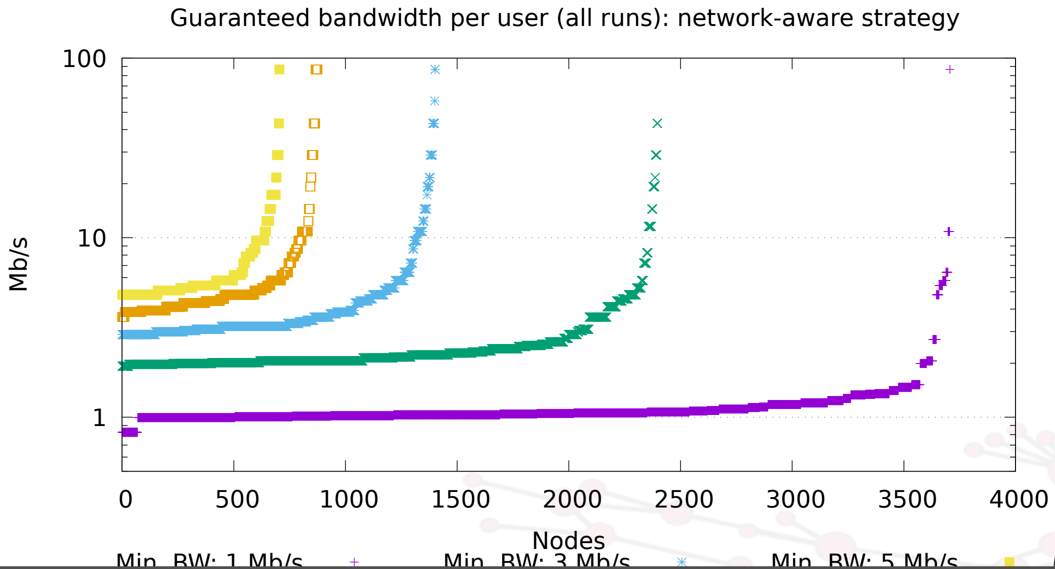
Credits

- Code by myself, Gabriele Gemmi and Daniele Mazzetti (the web interface)
- Ideas and discussions by the researchers from netCommons (paper under review right now. . .)
- Co-Funded by the Horizon 2020 programme of the European Union, Grant Number 688768

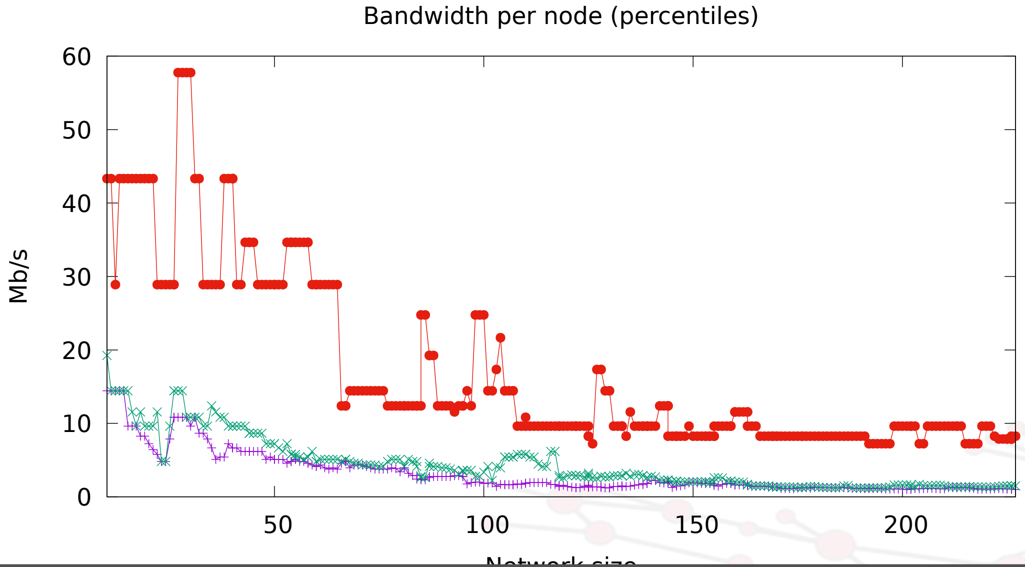
Bandwidth distribution (10 runs)



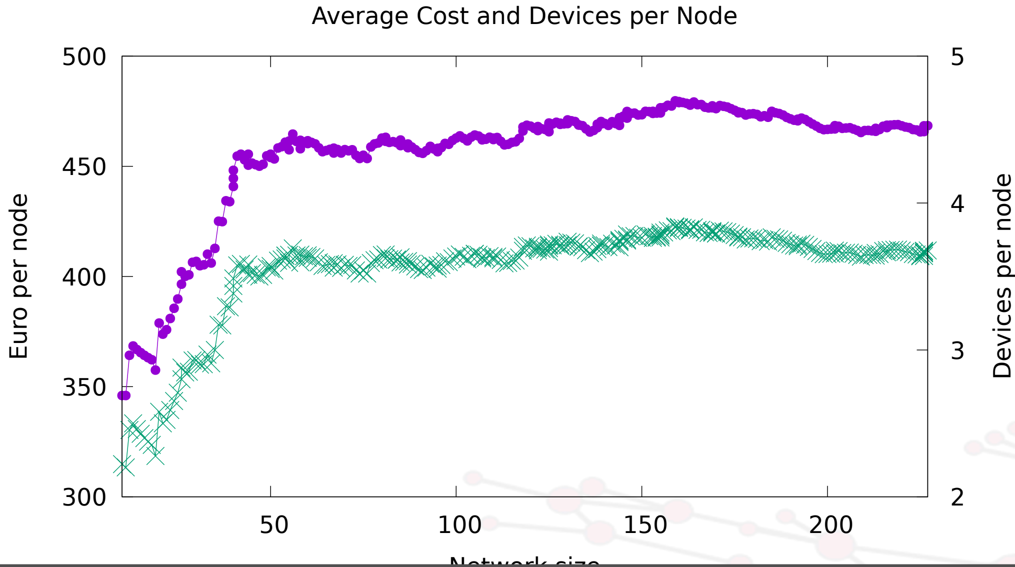
Bandwidth distribution: network-aware attachment



Growth of one network: Bandwidth



Growth of one network: Price



More things to do with TSUT: Networks Domain

1. Not only CAPEX, but estimate OPEX too
2. Different technologies: TVWS, 5G, IoT...
   - Ex.: 5G needs an extreme densification of the BS, uses mm wavelength, can we estimate coverage and cost?
   - Nokia proposed to use mesh networks backhaul[1].
   - How feasible is it? How much people we can reach with a mesh backhaul for 5G?

[1] Chen et. al. "*5G Self-Optimizing Wireless Mesh Backhaul A Proof-of-Concept Demo on Mesh Interconnected Small Cell Wireless Backhaul*" INFOCOM '15

What more: Interdisciplinary Research Domain

1. Study economic incentives: what is the best strategy to share the cost?
2. Include more open data from national surveys: current Internet coverage, average income, age, education... $\rightarrow$ try to forecast **who** is going to be served by this technology: is it going to serve only the already connected ones (young, educated, middle-to-high income)?

Cost sharing: two layers network

- So far we assumed every node owner pays the same: is it the correct way?
  - Pros: equal
  - Cons: if you can't afford it, your're out; probably unfair
- Reality suggest alternatives. In the Sarantaporo.gr community network, they use a different mode:
  - Two kinds of node: supernodes and leaf nodes
  - Supernode owners pays for their infrastructure, leaf nodes for network access
  - Leaf nodes pay fees to the supernode owners
- In a project deliverable (D2.8) we elaborated possible cost sharing strategies.

Cost sharing: introduce CNO

- In some cases, local heuristics are not enough
- One node needs more capacity to let other nodes connect, but the owner has no incentives to upgrade the hardware
- We could introduce a Community Network Owner, a collective body that suggests network improvements with a global view on the network evolution.
- CNO can collect money from node owners and invest some to “refactor” pieces of network
- Question: who should contribute to the CNO? how much?
- Potential Answer: central nodes are important for the network, should pay less. Peripheral nodes are freeriders, should pay more.
- **Main issue**: To test strategies, we need a demand model. . .

Warning!

What follows is a mix of half-baked ideas and some handswaving! 👋

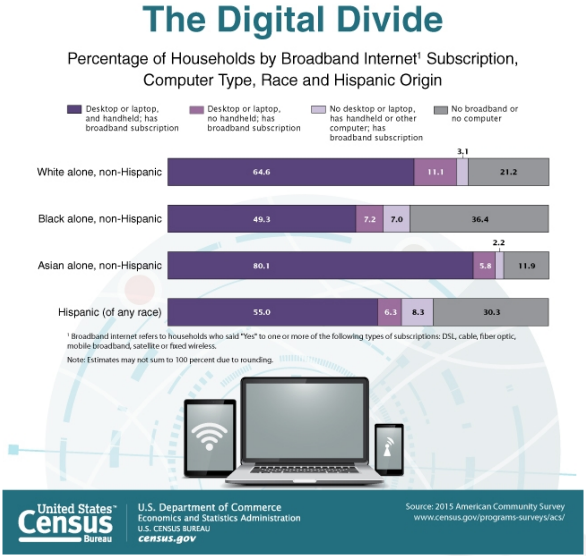
Nodes Generation

- So far, we pick new nodes at random.
- What if we use more open data to choose locations that are more or less feasible?
- National surveys publish huge open data sets with demographics: income, age, education
- These data sets are published down to the "block" detail
- Can we estimate the possible demand of connectivity based on those parameters?
- Can we compare the effectiveness of our cost sharing models with realistic demand constraints?
- Can we tune them based on the area (urban/suburban...)

Societal Impact

- If our mesh networks do not evolve only depending on geographic/terrain/technological constraints, who do they reach?
- Do they produce more or less inequality? Do they connect the already connected one?
- What about the other societal impact?

Digital Divide in the USA: 2018



Societal Impact

- How do mesh networks (or any other network we can model) compare, in terms of societal inclusion?
- The fact that we pose some technological constraints, introduces an intrinsic bias towards some social groups?
- Can we compare different technologies?

One last bit: Governance

- A distributed network grows “organically” and in an unplanned way
- It replaces a proper planning with redundancy obtained with network density
- The more it maintains its flat, unplanned organization, the more agile it remains, the easier it is to govern
- With lightweight nudging and consensus these networks grow up to hundreds of nodes