

# Towards Scalable Community Networks Topologies<sup>☆</sup>

Leonardo Maccari<sup>a</sup>, Gabriele Gemmi<sup>a</sup>, Renato Lo Cigno<sup>a</sup>, Merkourios Karaliopoulos<sup>b</sup>, Leandro Navarro<sup>c</sup>

<sup>a</sup>Department of Information Engineering and Computer Science, University of Trento, Italy

<sup>b</sup>Athens University of Economics and Business, Athens, Greece

<sup>c</sup>Department of Computer Architecture, Universitat Politècnica de Catalunya, Barcelona, Spain

---

## Abstract

Community Networks (CNs) are grassroots bottom-up initiatives that build local infrastructures, normally using Wi-Fi technology, to bring broadband networking in areas with inadequate offer of traditional infrastructures such as ADSL, FTTH or wide-band cellular (LTE, 5G). Albeit they normally operate as access networks to the Internet, CNs are ad-hoc networks that evolve based on local requirements and constraints, often including additional local services on top of Internet access. These networks grow in highly decentralized manner that radically deviates from the top-down network planning practiced in commercial mobile networks, depending, on the one hand, on the willingness of people to participate, and, on the other hand, on the feasibility of wireless links connecting the houses of potential participants with each other.

In this paper, we present a novel methodology and its implementation into an automated tool, which enables the exercise of (light) centralized control to the dynamic and otherwise spontaneous CN growth process. The goal of the methodology is influencing the choices to connect a new node to the CN so that it can grow with more balance and to a larger size. Input to our methodology are open source resources about the physical terrain of the CN deployment area, such as Open Street Map and very detailed (less than 1 m resolution) LIDAR-based data about buildings layout and height, as well as technical descriptions and pricing data about off-the-shelf networking devices that are made available by manufacturers. Data related to demographics can be easily added to refine the environment description. With these data at hand, the tool can estimate the technical and economic feasibility of adding new nodes to the CN and actively assist new CN users in selecting proper equipment and CN node(s) to connect with to improve the CN scalability.

We test our methodology in four different areas representing standard territorial characterization categories: urban, suburban, intermediate, and rural. In all four cases our tool shows that CNs scale to much larger size using the assisted, network-aware methodology when compared with de facto practices. Results also show that the CNs deployed with the assisted methodology are more balanced and have a lower per-node cost for the same per-node guaranteed bandwidth. Moreover, this is achieved with fewer devices per node, which means that the network is cheaper to build and easier to maintain.

**Keywords:** Community Networks, Stochastic Network Evolution, Economic Incentives, Bottom-Up Broadband, Alternative Architectures

---

## 1. Introduction

Community Networks (CNs) are grassroots bottom up networks often built as 802.11-based Wireless Mesh Networks (WMNs). CNs are flourishing in Europe and grow in many different environments worldwide, their “preferred” ecosystem being areas where, whatever the reason, standard telecommunication infrastructures do not work properly. Often, these are areas of “market failure,” i.e., areas where commercial operators deem it non-profitable to invest<sup>1</sup>. CNs also flourish in places with fervent cultural life, where people share strong community links and/or social/political ideals, and invest into a local infrastructure that can bestow on the community much more than the standard Internet in terms of digital divide reduction,

rich and non-commercial services, and local economy support. It is now acknowledged that CNs, even if they sometimes fail, form an integral part of the global Internet. As such, they should be nurtured by the regulatory system and policy makers, because when they have success, they serve as a catalyst for the socio-economic development and well being of their region<sup>2</sup>.

A CN is typically launched thanks to the initiative of a small group of people, whose motives may range all the way from enthusiasm about technology and do-it-yourself practices to social activism and political causes [2]. The group members invest personal resources (effort, time, money) to set up a first small set of network nodes that ensure connectivity to the rest of the Internet or can support the provision of local services. This initial burst of activity normally gives rise to a network of a few nodes and a topology that is mainly determined by the location of the group members’ homes. Over a second longer phase, the network grows thanks to the addition of nodes by people who join the network

---

<sup>☆</sup>This work was financed partially by the European Commission, H2020-ICT-2015 Programme, Grant Number 688768 “netCommons” (Network Infrastructure as Commons).

Email addresses: leonardo.maccari@unitn.it (Leonardo Maccari), gabriele.gemmi@stud.unitn.it (Gabriele Gemmi), renato.locigno@unitn.it (Renato Lo Cigno), mkaralio@aueb.gr (Merkourios Karaliopoulos), leandro@ac.upc.edu (Leandro Navarro)

<sup>1</sup>A condition that, according to ITU [1], applies to roughly 50% of the world population, and there is not indication that this figure is actually decreasing.

<sup>2</sup>For additional information on CNs, their diffusion in the world, their size and characteristics see the web page of the netCommons project; in particular, refer to Deliverable 1.2 “Report on the Existing CNs and their Organization,” and Deliverable 1.4 “Report on the Governance Instruments and their Application to CNs.”

and become members of the community. The network growth during this second phase is a distributed process with a strong crowdsourcing flavor that clearly distinguishes it from the top-down planning practised in conventional communication networks. The existing network nodes that become points of network attachment for new nodes that join the CN are typically determined *locally* and *heuristically*, depending on the node geo-location, the CN coverage in the area, as well as the availability and cost of proper hardware devices. Since the cost of the added node is normally sustained by the new CN member, the decision tends to be myopic and “greedy,” in that it only seeks to reduce the cost the new member incurs.

These *local* decisions, however, *do* shape the process of network evolution. They determine the main *global* properties of the resulting network topology, such as the average length of the shortest paths to the Internet gateway(s), the robustness of the network to topology failures, as well as the distribution of its overall capacity and traffic load across its nodes and links. Hence, they strongly influence the network performance and dictate the overall cost of the developed infrastructure.

The empirical analysis of CN topologies that evolve guided by fully decentralized decisions, without any central coordination or intervention, has provided evidence of pathologies and emergent risks. These include high dependence (in terms of connectivity and routing functionality) on a single or a few nodes, which may turn to single points of failure for the CN (if, for some reason, their owners lose interest in the CN or move to another place), and large differences in routes and speeds connecting end users to the Internet [3, 4, 5].

A key question arising in this context is whether it is possible to intervene in the CN growth process in order to steer its topology towards patterns that better serve its robustness and sustainability. Note that this question is distinctly different than the one faced by commercial network operators who plan their network as a top-down process. The CN infrastructure develops sequentially, in response to the time series of “join” events by end users, and there is no control over these events, i.e., *where* new nodes are to be installed. The remaining issue, then, is to what extent one can influence *how* these nodes are added, i.e., how they connect to the CN. This can only be obtained by influencing or constraining the local choices of the new users when they set up their own nodes.

The first contribution of our work consists in showing that simple algorithms can drive these local decisions and lead to drastic network performance improvements. We formulate the problem mapping it to the max-min fair routing problem (e.g., [6][7]) and propose a greedy heuristic to solve it. We show that this greedy heuristic, bringing a network-wide view into the CN evolution process, helps the CN scale up to several hundreds of nodes and 2–3 times the size it would grow when the addition of the nodes is driven only by the new member’s cost minimization.

The second contribution of this work, which serves as an enabler for the first contribution, is a tool that can simulate and assess growth strategies of a CN exploiting (very) detailed topological descriptions of the CN deployment area and economic and technological constraints to set up the CN nodes. The topological description leverages open source data from OpenStreetMap<sup>3</sup> and LIDAR-based estimates

of building heights with an horizontal and vertical precision better than 1 m, and feed appropriate propagation models to infer the availability and quality of wireless links between pairs of CN nodes. Results with this tool are obtained for four areas with different population density and topological features, which are representative of four classes of the regions’ territorial characterization of the OECD (Organization for Economic Co-operation and Development)<sup>4</sup>.

The source code of this tool is published as an open source project and it is available on-line<sup>5</sup>. We highlight that the tool can be used in two main different ways. The first one, hinted above, is like a planning tool to help the management of a CN taking informed and rational decisions on the network expansion. The second one, maybe even more interesting, and how we use it in this paper, is to use it as a feasibility analyzer: Given an area or region, what is the potential to build a CN? And what is the probability that the CN can grow up to a sustainable dimension?

## 2. Background and System Model

The deployment of CNs is a participatory and evolutionary process. The network grows over time as new users join the community network and new nodes are added to existing ones.

### 2.1. CN node and installation cost budget

In a CN, as well as in our simulator, a “node” is an installation of a wireless mesh node and includes an indoor part and an outdoor part. The former is a wireless router, which is configured to redistribute Internet access in the user’s house and perform IP routing. The latter is a composition of various elements, some of which generate a fixed cost and some a variable cost. The fixed cost counts a metallic pole needed to elevate the wireless devices a few meters above the roof level. We consider a maximum height of 4 m for such a pole when mounted on a standard CN node, whereas this height may go up to 10 m for gateway nodes, which attach to the Internet. Another cost item is the cable (generally one Ethernet cable, or one Ethernet plus power cable), which feeds the outdoor devices with power and connects them to the indoor devices. On the roof, one can place a PoE (Power over Ethernet) switch, which splits power and data to a number of outdoor wireless devices. These are full-fledged wireless routers with CPU, memory, radio module and embedded antenna, and we refer to them as “devices” for brevity. These devices make up the variable part of the cost. The maximum number of devices per node (gateway) is set to four (ten), which is a reasonable number given the power consumption and the physical space available on the pole.

We set the fixed cost to 200 €, a somewhat arbitrary number that changes from one country/location to another, but which is reasonable from the experience we have in Italy, Spain and other EU countries. Being constant its impact on the results is marginal. The variable cost depends on the number and the type of devices according to the prices we collected from websites of wireless device re-sellers (see Table 1). Different CNs may use different node designs; the one we propose, and its implications on the cost, draws on the hands-on experience the authors had with existing CNs.

<sup>3</sup>See [https://wiki.openstreetmap.org/wiki/API\\_v0.6](https://wiki.openstreetmap.org/wiki/API_v0.6)

<sup>4</sup>See the OECD OECD web site for further details.

<sup>5</sup>See <https://github.com/AdvancedNetworkingSystems/>

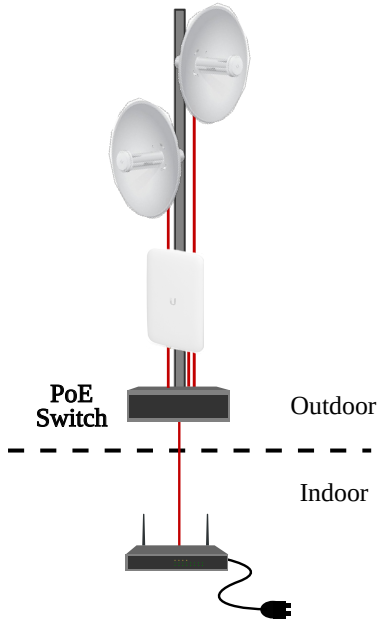


Figure 1: The components of a wireless mesh node.

## 2.2. Actors and roles

There are typically two main actors in community networks. The first one is the small group of people who lead the initiative and set up the first wireless nodes. More often than not, they remain involved in the CN, undertaking a major role in its maintenance and management. The need to interact with external entities (e.g., municipalities, policy makers, regulating authorities) motivates the organization into various types of legal entities, varying from associations and cooperatives to non-profit (or, rarely, small for-profit) ISPs. We refer to the legal entity managing the CN as the CN Operator (CNO), regardless of its legal status.

The second main actor in CNs are the end users who join the network by contributing their own equipment, money and effort. At the same time, through the nodes they add, they expand the geographic coverage of the CN, making it accessible to more people. This participatory network deployment process challenges sustainability in several ways, whether this is approached from techno-economical only or a broader socio-political point of view [2]. In this paper, we are concerned with the first aspect. In particular, we focus on the challenge of maintaining a sustainable network topology as the CN scales up with the addition of new users. Since CNs grow bottom-up in response to community interest in them, end users have a direct impact on the CN topology and coverage, as these evolve over months and years.

The new node installation process is often in the hands of, or at least assisted by, the CNO team. Therefore, the CNO could give recommendations or even constraints to the users on which node to connect to, or which device to use. It may also request a user to add a new device to his/her node. However, it can not realistically trigger a set of coordinated changes on many nodes to strongly modify the network topology.

## 2.3. CN deployment: an evolutionary participatory process

The deployment of a CN starts with a small number of nodes, then grows based on end-user requests.

### 2.3.1. Set-up of the first CN nodes by the CNO

Generally, the CNO deploys an initial set of wireless nodes  $N_0$ . The location of these nodes coincides with the houses/residences of the CNO team members or friends of theirs. Their selection may be optimized to maximize the aggregate geographic coverage. Among these nodes there is one or more CN gateway(s) attached to the Internet with a broadband connection. The set-up of these nodes yields an original CN topology graph  $G_0 = (V_0, E_0)$ , with  $|V_0| = N_0$ . In this work, without loss of generality, we assume that  $N_0 = 1$ . We also assume that the gateway has sufficient uplink capacity for the whole network; hence, in general, this is not a standard ADSL connection but an appropriate gateway set up by the CNO with traditional operators or even directly in an Internet Exchange.

### 2.3.2. Evolutionary growth of the CN and CNO interventions

The second and main phase in the CN growth process is driven by potential community members who submit requests to join the CN. In doing so, the users incur a cost  $C_n$  to set up a node at their home. On the other hand, the deployment of a new node  $n$  also implies a cost  $C_{n,l}$  for setting up the peer point for the link  $l$  at the point of attachment to the CN. This cost is closely related to the hardware chosen for the node, which in turns depends on the distance and the quality of link  $l$ . For example, a cheaper device with smaller wireless range might suffice for attaching to the closest node, whereas a more expensive device with higher range would be needed to reach a more distant one, which may lie closer to the Internet gateway.

Several criteria may apply when choosing the point of attachment to the CN. If this choice is not subject to some form of control/regulation on the CNO side, a new user might end up connecting to the CN node that can be received most powerfully and/or lies closer to the user. This is a selfish choice that myopically tries to maximize the quality of the local radio connection and minimize the cost incurred by the user (cost of purchased device). In our approach, we let the CNO intervene in this choice by choosing the point of attachment to the CN among the available alternatives.

Any time a new node is added to the CN, the CNO tries to ensure that, after the new addition, all CN nodes can obtain a minimum acceptable share of the network capacity. The CNO may terminate the CN growth process if it figures out that the addition of new nodes results in unacceptable network performance degradation.

Once they enter the network, users may pay a recurring fee to contribute to the Internet interconnection and network maintenance costs, i.e., the CN Operating Expenditure (OPEX). In this paper we are interested in how much such a network can grow, and how much it can cost, so we limit our analysis to the Capital Expenditure, i.e., the initial cost to deploy the CN infrastructure, assuming that OPEX remains balanced and it is generally small being CNs participatory enterprises.

## 3. Controlling the CN Growth Process

Consider the onset a CN beginning at time  $t_0$ , when the first CN nodes are set up by the CNO. Among the present nodes there is

a gateway  $g$ . We would like to track the evolution of the CN up to the time  $T$ , when the CNO terminates the CN growth process because the minimum per-node bandwidth  $R_{min}$  toward the Internet falls below a given threshold, say  $R_{thr}$ , for a fraction of nodes  $p$ .

### 3.1. Problem Formulation

Requests for the addition of new nodes to the CN come at random times  $\{t_j\}$ ,  $t_j \in [0, T]$ ,  $j \in \mathcal{N}$ , with  $\mathcal{N}$  the set of users who eventually submit a request to join the CN. This set is bounded by the number of potential users (typically buildings in the area), but can be larger than the actual number of nodes when the CN terminate its growth, as not all join requests might be fulfilled.

Every time a new end user pose a request to add node  $n$  at time  $t_n$ , the following actions are completed.

- The CNO checks that node  $n$  can be connected to one of the CN nodes that lie within its range. Call  $\mathcal{L}_n$  the set of links that identify nodes reachable by  $n$ , each with different worst-case rates depending on the link quality (distance and propagation environment), the used devices installed, and the current CN topology.
- If  $n$  cannot be connected to the network, it is added to a list  $V_c$  of candidate nodes. When another new node  $m$  is added to the network the CNO will check if any of the candidate nodes can be added through  $m$ .
- If the node  $n$  is connected, normally with a single device to begin with, it becomes a point of attachment for more nodes that could join the network (in this case probably adding to  $n$  more devices, up to the limit of four we have defined in Section 2.3.2), either those in the candidate list, or additional, currently unknown, nodes.
- For every new node added, the CNO incurs a cost  $C_{n,l}$  that depends on the kind and number of devices necessary to setup the link  $l \in \mathcal{L}_n$  selected. The node feeds the network with its traffic demand, which has to be served by the network, possibly reducing the available bandwidth for the demands of other users.
- The CNO checks whether the preset performance target  $R_{thr}$  is not preserved for a fraction  $p$  of nodes after the addition of the new node. If not, the network evolution stops. In practice, upon such events, the CNO would need to revisit the CN topology and consider upgrades that could allow more users to join the CN. However, to set a concrete context for comparing our approach to alternatives, we consider the evolution of the CN topology up to the time  $T$  that the performance criterion is violated for first time, halting the further addition of nodes.

The evaluation of the minimum bandwidth  $R_{min}$  available for any node is a non trivial task, and it depends on the actual routing protocol adopted by the CNO, as we describe in the next section.

### 3.2. Estimation of network growth

Ideally, the CNO would like to coordinate the choice of new links so that its growth stopping time  $T$  is as high as possible; namely, the CN can scale up to the maximum possible size while providing all

nodes with a minimum worst-case throughput. With reference to the sequence of join request times  $\{t_j\}$ , this is essentially a dynamic optimization problem. However, without prior knowledge, even probabilistic, about the sequence of new user join requests, finding the optimal link addition policy that maximizes  $T$  is practically impossible.

In the absence of any knowledge about the sequence of join events, a plausible approach to the optimization problem is a procedure that any time a new user joins the CN, the CNO chooses the link that maximizes the worst-case per node throughput given the topology of the CN at time  $t$ .

More specifically, let  $G_t = (V_t, E_t)$  be the directed graph describing the CN topology at time  $t = t_n$ , when the request to add node  $n$  arrives, and  $\mathcal{L}_n$  the links that are feasible between  $n$  and CN nodes  $n' \in V_n$  within its useful range, i.e., its potential neighbors. For each link  $l \in \mathcal{L}_n$ , the CNO hypothetically constructs the topology  $G_t(l) = (V_t \cup n, E_t \cup l)$  and computes the best minimum worst-case rate  $R_{min}(l)$  that can be made available across all nodes  $i \in V_t \cup n$ . To find this, it could solve an instance of the max-min fair routing problem with unsplittable demands [6, 7] over the  $G_t(l)$  graph, which features one source (gateway node) and many sinks (all other CN nodes). The problem solution would yield an approximately-optimal routing configuration, i.e., single-path routes from the gateway to all CN nodes, that maximizes the minimum worst-case throughput across all feasible routing configurations.

The CNO would then pick up the link  $l_n$  that results in the highest value of  $R_{min}(l)$  across all links  $l \in \mathcal{L}_n$ :

$$l_n = \operatorname{argmax}_{l \in \mathcal{L}_n} (R_{min}(l)) \quad (1)$$

as the one to set up between  $n$  and the CN.

In practice, several factors undermine the practical relevance of such a systematic approach. First, the radio interference considerations in the wireless multihop setting of a CN add significant extra complexity (see for instance [8]) to the already NP-hard max-min fair routing problem. The use of directional links may weaken but not fully cancel the impact of radio interference. Second, the (approximately) optimal solution would be practically achieved only if the combination of the link scheduling function at the wireless routers with the flow control function at the end points produced max-min fair rate allocations for any routing, which is definitely not the case with state of the art IP-based routing and congestion control protocols.

### 3.3. Link selection algorithm

The capabilities of real protocols, with specific reference to the Optimized Link State Protocol (OLSRv2) [9] can be summarized by the following points.

- Each node is able to estimate the available link bandwidth. Modern routing protocols communicate with the wireless devices to obtain the negotiated bit-rate of the last sent packet, and use this data as a link metric [10].
- Each node is able to estimate the number of links that are directly interfering with the device on the receiving side. In the simplest case, this is the number of ingoing links to the destination device, which can be communicated by the

neighbor node. There are protocols that use a similar approach to penalize the use of links that share the same channel with more than one neighbor<sup>6</sup>.

- It is not possible to perform a centralized optimization based on the instantaneous knowledge of the link occupation. Every node estimates the available bandwidth per link at steady state conditions and shares this knowledge with the other nodes. Routing is performed based on this knowledge, which can be partially outdated.
- No multipath routing is supported, so that a flow cannot be split among equivalent shortest paths.

In light of the general constraints above, we can now formulate two link selection strategies that can be implemented in our tool when a new node  $n$  join request is considered. The first one is a network aware strategy that, with acceptable computational complexity, grants that the CN growth is not hampered by local myopic strategies. The second one is instead a local, myopic strategy that we use as a baseline for comparison.

### 3.3.1. Network aware strategy

This strategy goal is selecting the link  $l_t \in \mathcal{L}_n$ , and hence the neighbor  $n' \in V_n$ , that guarantees the maximum minimum bandwidth from any node in the network to the gateway  $g$  after the node  $n$  is added to the CN. Compared to Eq. (1) this strategy still finds a time-dependent local optimum conditioned on the sequence of request arrivals  $\{t_j\}$ , but instead of computing the optimal global routing and rate allocation, is selects the solution that yields the best minimum bandwidth given the routing protocol adopted by the CN.

As a first step, if  $\mathcal{L}_n$  is not empty, for each link  $l_t \in \mathcal{L}_n$  a new topology  $G_t(l_t) = (V_t \cup n, E_t \cup l_t)$  is considered, and every directed link  $l = (h, k)$ ,  $h, k \in V_t \cup n$  of the network is assigned a cost:

$$l.metric = \frac{l.sharing\_factor}{l.max\_bitrate} \quad (2)$$

where  $l.max\_bitrate$  is the maximum bit rate achievable by the device in  $k$  where link  $l$  terminates,  $l.sharing\_factor$  is the number of links terminated in  $k$  that share the same device with  $l$ . The topology  $G_t(l_t)$  annotated with  $l.metric$  is a weighted graph where Dijkstra algorithm can be used to compute minimum cost paths  $path(h, g)$  from each node  $h$  toward the gateway  $g$ .

Second, knowing all the minimum-cost paths, we can compute the number of paths directed to  $g$  that are active on every link  $l$ :  $l.paths\_sharing\_factor$ . The bandwidth that is available for a path in each link  $l$  is equal to the ratio of the nominal link capacity,  $\frac{1}{l.metric}$  (Eq. (2)), divided by the number of active paths toward  $g$  on the link  $l.paths\_sharing\_factor$ , assuming equal sharing of the link capacity between all  $path(h, g)$  insisting on  $l$ . Hence, the minimum available bandwidth for each node  $h$  corresponds to the minimum bandwidth available over all links in  $path(h, g)$ :

$$minBW(h) = \min_{l \in path(h, g)} \left[ \frac{1}{l.metric \cdot l.paths\_sharing\_factor} \right] \quad (3)$$

<sup>6</sup>See for instance the Batman-adv multi-link optimization <https://www.open-mesh.org/projects/batman-adv/wiki/Network-wide-multi-link-optimization>

Third, we need to select the link  $l_t \in \mathcal{L}_n$  that maximized  $minBW(h)$  over all nodes and all possible  $G_t(l_t) = (V_t \cup n, E_t \cup l_t)$ :

$$l_n = \operatorname{argmax}_{l_t \in \mathcal{L}_n} \left[ \min_{h \in V_t \cup n} (minBW(h)) \right] \quad (4)$$

The complexity of this search is polynomial as it requires to enumerate all the minimum weight paths computed using Dijkstra on every edge in the networks, which is acceptable both for “what if” analysis before starting a CN, and for run time (of the CN development) addition of single requests.

### 3.3.2. Local strategy

This strategy simply selects the link  $l_t \in \mathcal{L}_n$ , and hence the neighbor  $n' \in V_n$ , that maximizes the local bandwidth between  $n$  and  $n'$ . Formally:

$$l_n = \operatorname{argmax}_{l_t \in \mathcal{L}_n} \left( \frac{1.0}{l.metric} \right) \quad (5)$$

and, in case of tie the solution that requires a lower cost  $C_{n,l}$ .

### 3.4. Terminating the CN growth process

The implementation of the stop criterion is straightforward. After a new node is added, we check whether the fraction of nodes with  $minBW(n)$  below the threshold  $R_{thr}$  is higher than a target *percentile*. If this is the case, the CN growth process is terminated. Equation (3) is a worst-case, peak allocation criterion with multiplexing factor equal to 1, and not a contractual SLA (Service Level Agreement); thus building a network where it is satisfied for, say 80-90% of the nodes is actually very safe and ensures that the performance of the network is satisfactory for any user in any operation conditions with high probability.

## 4. A CN Planning Tool

So far we have described what a tool to help in the development and deployment of CNs may work, but we need to describe the tool and understand how link selection strategies can be integrated in this tool. As we mentioned, the tool can be used to assist in the deployment of CNs, simply using it with the real flow of join requests, or can be used as an emulator trying to understand if a CN is feasible given an area of potential interest. Clearly, CNs may take months or years to grow, so that its first use is difficult. In this work, we describe and use it as an CN growth emulator, or simply a *generator*.

The output of an emulation run is a topology  $G_T = (V_T, E_T)$ , where the pedix  $T$  indicates that the possible growth of the CN has “Terminated.” The topologies produced are annotated with estimates of the available bandwidth in each link and the total amount of money invested in the CN deployment. Different metrics can then be used to assess critical features of those topologies, such as the expected throughput they provide to end users and their resilience to failures, or simply the dimension the CN has reached before the stopping condition in Section 3.4 was met.

The generator implements a greedy approach (meaning that each join request is considered in order and isolation from the others). Its execution and temporal evolution is described by Algorithm 1 (commented later) and consists of four main components.

1. A stochastic engine that implements the sequence of requests  $\{t_j\}$  based on some criteria that depends on the area selected for the installment of the CN;
2. A database of open data. This database includes building shapes and altitude, and it is used to implement the `getLoS` function in Algorithm 1.
3. Different link selection strategies (the `getBestNeighbor` routine in Algorithm 1) and CN growth stopping criteria (the `checkStopCondition` routine in Algorithm 1). The two link selection strategies defined in Section 3.3 and the growth stopping criterion defined in Section 3.4 are currently implemented in our tool.
4. Simple metrics that are used to assess a-posteriori the CN topological properties.

Each component is easily extensible to support other strategies, and contains several parameters that can be tuned to reproduce a wide range of real world conditions.

#### 4.1. The stochastic engine

The driver of our tool when used as a feasibility analyzer is fundamentally an event generator that select the sequence  $\{n, t_n\}$  and drives the temporal evolution described in Algorithm 1, similarly to any event-driven simulator. If the tool is instead used as a planning tool, the driver is a real trace of requests  $\{n, t_n\}$  leading to a trace-driven simulation.

As in any event driven execution Algorithm 1 is fundamentally an infinite loop until an ending condition is met (line 3). The stochastic part of it is `getNewRequest()` at line 4, that selects a new potential node to join the CN. What we have implemented so far is a simple random selection among all the buildings not yet connected, complemented with the browsing of all the nodes that have requested to join, but were not in LoS with any node already present in the CN and have been memorized in the  $V_{wait}$  set. These simple driver can be extended to include demographics of the area, starting from the number of families in a building, down to census data (education, income, ...) that may affect the inclination toward joining a CN.

#### 4.2. The open data database

The CN topology generator collects data from three sources and organize them into a database for efficient use during the execution. We remark that all data we collect are Open Data, thus the tool can be freely used by anyone.

1. Street maps including building shapes, extracted from OpenStreetMap (OSM) and other public open data repositories. Whereas OSM is generally very precise in urban areas, open data sets from public administrations tend to be more precise, though less up-to-date, in rural areas. In some cases (e.g., in France) the public open data sets have already been imported in OSM.
2. Building altitude profiles obtained from LIDAR (Light Detection and Ranging) traces. Several public bodies have published open data from LIDAR survey campaigns, with

---

#### Algorithm 1 Greedy approach to CN topology control

---

**Input:** The gateway node  $g$ , bandwidth threshold  $R_{thr}$ , tolerable fraction of nodes below threshold  $p$ ,

**Output:** A feasible network topology  $G_T = (V_T, E_T)$  and the total cost  $C$  for its deployment

```

1:  $C=0, V_t=\{g\}, E_t=\{\}, V_c=\{\}$ 
2:  $cont=0$ 
3: while  $cont \leq p \times |V_t|$  do
4:    $n = \text{getNewRequest}(V_c)$ 
5:   if  $n == NULL$  then
6:     break
7:   end if
8:    $V_n = \text{getLoS}(V_t, n)$ 
9:   if  $V_n$  is empty then
10:     $V_c.append(n)$  // populate the candidate list
11:   else
12:     $n' = \text{getBestNeighbor}(n, V_n, V_t, E_t)$ 
13:     $l = \text{newLink}(n, n', G_t)$ 
14:     $C = C + \text{cost}(l, V_t, E_t)$ 
15:     $V_t = V_t \cup n$ 
16:     $E_t = E_t \cup l$ 
17:     $cont, \_ = \text{checkStopCondition}(V_t, E_t)$ 
18:   end if
19: end while
20: return  $G_T(V_t, E_t), C$ 
21:
22: Function getNewRequest() // return a new node that requests
    to join the CN or NULL if there are no more nodes. Can re-use
    nodes in  $V_c$  if the network has grown since last addition to  $V_c$ .
23: Function getLoS( $V_t, n$ ) // returns the set of nodes in  $V_t$  that
    are in LoS with  $n$ 
24: Function cost( $l, V_t, E_t$ ) // cost for the addition of link  $l$ 
25: Function checkStopCondition( $G$ ) //See Algorithm 4
26: Function getBestNeighbor( $G$ ) //See Algorithms 2 and 3

```

---

various level of precision<sup>7</sup>. Public administrations in several countries publish much more precise data sets; the one we use reaches a precision of one point per squared meter.

3. Technical specifications of real 5G-ready (802.11ac) devices from the Ubiquiti equipment manufacturer. For each device, the database stores the maximum transmission power, the antenna gain and aperture, the sensitivity for each supported Modulation and Coding Scheme (MCS), and an average price, extracted from official stores (dated Sept. 2018). Table 1 reports the features of the devices we have used in the simulations in Section 5.

All sources of data can be extended. The technical specifications in particular can be extended to any vendor and any device depending on the commercial choices of the CN. For the purpose of this work we obviously don't have a commercial strategy are interested

---

<sup>7</sup>The Shuttle Radar Topography Mission: <https://www2.jpl.nasa.gov/srtm/> is a public repository of terrain elevation profiles, but its precision does not allow to estimate building heights (roughly one point every  $900m^2$ ).

Name	Avg. Price (EUR)	Beamwidth Angle (H,V degrees °)	Sensitivity (dBm)	Max TX Power (dBm)	Antenna Gain (dB)	Max distance (km)
ISO90	200	90,30	-65	21	14	1.34
ISO45	112	45,45	-65	21	14	1.34
LB	73	20,10	-65	21	23	3.79
NB	100	30,30	-65	20	19	2.39
NS	134	60,20	-65	21	16	1.69
NSL	49	50,40	-65	21	13	1.20
PB3	110	20,10	-65	21	22	3.38
PB4	129	20,10	-65	21	25	4.77
PB5	185	20,10	-65	18	27	6.00

Table 1: The technical specifications of the Ubiquiti 5G-ready devices included in the simulator, limited to the highest Modulation and Coding Scheme (MCS9).

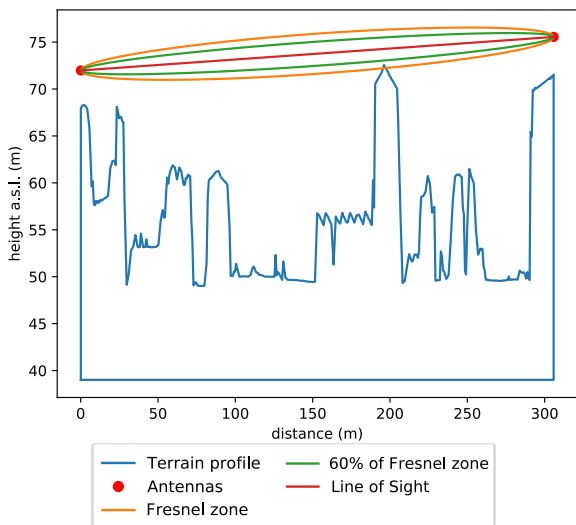


Figure 2: The altitude profile between two random buildings in the city of Florence, and the computed line of sight and Fresnel zone.

to explore the final cost of deployment in relative terms and not absolute ones, so devices from a single vendor are a good enough choice.

With these data we can verify if, placing devices on the roofs of any two buildings (each network node corresponds to a building), we can expect to achieve line of sight (LoS) between them. If this is the case, we compute the theoretical path loss with an attenuation exponent set to two, and then we refine it through a single knife edge approximation model, which takes into account the occupation of the Fresnel zone [11]. Figure 2 shows the result of the process of computation of the line of sight between two buildings in the city of Florence. The two antennas are placed 4 meters above the building roof, the red line is the LoS (which is available, in the case of these two buildings), the orange line is the computed Fresnel zone, and the green line contains 60% of the area of the Fresnel zone. Using the mentioned single knife edge approximation we are able to estimate a realistic value for the path loss on this potential link.

We assume each device is configured in ad-hoc mode (or equivalent) and we apply the European regulatory limits that set an upper bound of 30 dBm to the emitted power in the unlicensed band around 5 GHz. Using the antenna gain and the sensitivity values reported in the device data sheet, we estimate the MCS and

thus the negotiated bandwidth of the link.

A typical problem in mesh networks is how to assign channels to links in order to avoid interference with neighbouring nodes. The problem can be described as a variant of a graph colouring problem and it is NP-hard in the generic case [12], however, with the use of directional antenna this problem remains marginal.

#### 4.3. Link selection and growth stop

Each node in the CN topology is mounted on a building; the first node of the network is the gateway node as evident in line 1 of Algorithm 1. This node is chosen with care in order to avoid pathological conditions, as it happens in real cases. We typically choose a building that has reasonable connectivity such as the City/Town Hall, a hospital, or a university and we assume the availability of a 10m tralis structure that can be used to mount up to eight wireless devices.

The two link selection strategies defined in Sections 3.3.1 and 3.3.2 are implemented as described in the next two subsections. Note that, with either strategy, if every time we add a node we connect it to one single neighbor, the CN would become a tree. Instead, when we add a device to a node we always check if the new device can generate a connection with any node other than the intended one (on the same channel). After identifying the best neighbor  $n'$ , we estimate the cone periphery from the devices' datasheets extracting the angle at which the device exhibits 3dB loss and considering that as the real antenna aperture, next we check how many links can be generated with other existing nodes through the same device, and we allow a maximum of three outgoing links per device in order to limit the number of links that share a given device. Yet this limit is only lousily enforced, if  $n$  needs to connect to an existing device of  $n'$  that saturated the number of outgoing links, we allow to increase this number to let the new node enter the network.

##### 4.3.1. Local strategy

The strategy is described formally in Algorithm 2. For each node  $n'$  in the set of nodes  $V_p$  within LoS of  $n$ , the algorithm first computes the expected path loss and finds the two devices that need to be added to  $n$  and  $n'$  to obtain the maximum possible negotiated bit rate. It then picks the node  $n' \in V_p$  that guarantees the highest bit rate. In case  $n'$  already has a device pointing towards  $n$ , a new device is added only to  $n$  and is tuned on the same channel. Otherwise, a new device is also added to  $n'$  operating on a channel

---

**Algorithm 2** getBestNeighbor: local strategy

---

**Input:** new node  $n$ , set of feasible neighbors  $V_n$  for  $n$ , set of present nodes  $V_t$ , set of edges  $E_t$  (ignore other parameters)

**Output:** Neighbor that maximizes the bandwidth of the new link

```
1:  $bestNeigh = NULL$ ,  $linkBw = 0$ ,  $linkCost = inf$ 
2: for  $n'$  in  $V_p$  do
3:    $l = newLink(n, n')$ 
4:    $c = cost(l, V_t, E_t)$ 
5:   if  $l.bandwidth > linkBw$  OR
      ( $l.bandwidth == linkBw$  AND  $linkCost < c$ ) then
6:      $linkBw = l.bandwidth$ 
7:      $linkCost = c$ 
8:      $bestNeigh = n'$ 
9:   end if
10: end for
11: return( $bestNeigh$ )
```

---

---

**Algorithm 3** getBestNeighbor: network aware

---

**Input:** new node  $n$ , set of feasible neighbors  $V_n$  for  $n$ , set of present nodes  $V_t$ , set of edges  $E_t$ ,  $R_{thr}$ ,  $p$ ,  $g$

**Output:** Neighbor that minimizes the network-wide performance drop due to the new node bandwidth requirements

```
1:  $R = []$ 
2: for  $n'$  in  $V_p$  do
3:    $l = newLink(n, n')$ 
4:    $V'_t = V_t \cup n$ 
5:    $E'_t = E_t \cup l$ 
6:    $\rightarrow R_{min} = checkStopCondition(V'_t, E'_t, R_{thr}, p, g)$ 
7:    $R[n'] = R_{min}$ 
8: end for
9:  $bestNeigh = NULL$ 
10:  $bestR = 0$ 
11: for  $n'$  in  $R$  do
12:   if  $bestR < R[n']$  then
13:      $bestNeigh = n'$ ,  $bestR = R[n']$ 
14:   end if
15: end for
16: return( $bestNeigh$ )
17:
18: Function  $newLink(n, n')$ 
19: return a structure that contains the information of a new link
      (loss and bandwidth) between  $n$  and  $n'$ .
```

---

that is not already used by any device on  $n'$ . The local strategy, as its name suggests, tries to maximize the bandwidth available to the new node on the last hop.

This is a selfish strategy that does not take into account the impact of the new node addition on the rest of the CN, but it is often used since uses only local information and does not require any computation. It tends to be the de facto approach to CN expansion, thus we use it as baseline.

#### 4.3.2. Network aware strategy

The strategy is described formally in Algorithm 3. The network-aware strategy does not necessarily add the fastest new

---

**Algorithm 4** checkStopCondition

---

**Input:**  $V, E, R_{thr}, p, g$

**Output:** Nodes below threshold, lowest guaranteed bandwidth

```
1:  $path\_dict = []$ 
2:  $link\_bandwidth = []$ 
3: for  $l$  in  $E$  do
4:    $l.metric = l.sharing\_factor/l.max\_bandwidth$ 
5:    $l.paths\_sharing\_factor = 1$ 
6: end for
7: for  $n$  in  $V$  do
8:    $path = computeDijkstra(n, g)$ 
9:    $path\_dict[n] = path$ 
10:  for  $l$  in  $path$  do
11:     $l.paths\_sharing\_factor += 1$ 
12:  end for
13: end for
14:  $bottleneck = list()$ 
15:  $nodes\_below\_threshold = 0$ 
16: for  $l$  in  $E$  do
17:    $link\_bandwidth[l] = 1.0/(l.metric * l.paths\_sharing\_factor)$ 
18: end for
19: for  $n$  in  $V$  do
20:    $R_{min} = \min(\{link\_bandwidth[l] \text{ for } l \text{ in } path\_dict[n]\})$ 
21:    $bottleneck.append(R_{min})$ 
22:   if  $R_{min} < R_{thr}$  then
23:      $nodes\_below\_threshold += 1$ 
24:   end if
25: end for
26: return( $nodes\_below\_threshold$ ,  $\min(bottleneck)$ )
```

---

link. It rather evaluates all the candidate neighbor nodes as to how a link to them impacts the performance of the whole network (through the `checkStopCondition` at line 6 in Algorithm 3 that returns  $R_{min}$ ) and then selecting  $n'$  maximizing this value (lines 11-15). The fastest link is used as a fallback node selection criterion only if there is a tie.

#### 4.3.3. Topology growth stop criterion

Irrespective of how the neighbors of a new CN node are chosen, at some point the total demand will exceed the aggregate network capacity, and the network should stop accepting new nodes. The stopping criterion in our CN topology generator is described in Section 3.4, i.e., when the minimum available bandwidth estimated with Eq. (3) falls below  $R_{thr}$  for a certain percentage of nodes; Algorithm 4 formally describes its implementation. The absolute number of such nodes is computed in lines 22-14 and returned to the main loop in Algorithm 1. We assume that the bandwidth toward the Internet can be upgraded indefinitely, so that it never represent a bottleneck. This is clearly unrealistic, but let us estimate the intrinsic properties of the CN and not limitations due to commercial factors or policy decisions.

The network stops growing also in four more cases: i) there are no more node requests to join the CN; ii) none of the non-connected nodes have line of sight with any CN node; iii) non-connected nodes have line of sight with some CN node that already uses the



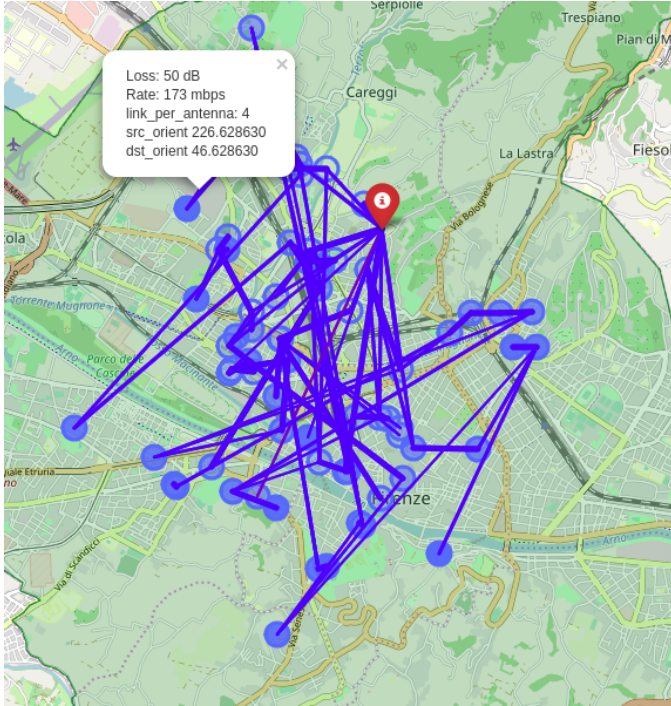


Figure 3: An example network realized on the area of Florence, for each link, the tool lets navigate its properties as shown in the pop-up.

maximum allowed number of devices, without any of these devices pointing to the right direction that would let a new node establish a link to it; iv) a mix of ii) and iii).

Figure 3 shows one example network realized using the data from the city of Florence (Italy), one of the areas we use to run our simulations, setting  $R_{thr}$  large so as to stop the growth with few nodes and obtain a readable picture.

#### 4.4. Graph metrics

Once a CN topology is generated, we compute two metrics over it:

**The network size:** The final number nodes with a path to the gateway by the time the stopping criterion is violated and no new nodes can be added;

**The average cost of a node:** Every time we add a link  $l$  from a new node  $n$  to an existing CN node  $n'$ , the total cost  $C$  of the network is increased by a fixed cost, plus the cost of the new devices that have to be purchased and installed (one, or two devices as explained in Section 4.3). The average value is key because it gives an estimation of the economical efficiency of the strategy, and it is more important than cost fluctuations from one node to another, considering costs are normally shared among various participants.

One might argue that as the CN grows, the overhead of the routing protocol may be a potential third metric. However, scalability studies for typical mesh routing protocols [13], such as OLSR, Babel, and BMX, show the negligible CPU (less than 10% load in a 10 € router [14]) and memory costs even for large mesh networks with 50-100 or more nodes. Network overhead is small for distance vector protocols

Scenario	Buildings	$km^2$	Buildings/ $km^2$
Urban (PU)	43853	102	429
Suburban (RIA)	6663	45	148
Intermediate (RI)	2052	34	60
Rural (RDP)	4414	182	24

Table 2: Number of building, dimension and building density in the four selected scenarios.

and it grows roughly linearly with the number of nodes (less than 200 bit/s for a 25 node network, and still less than 0.5-1 kbit/s for a 100 node network for a typical topology) while Link State protocols (like OLSRv2) generate a higher overhead. Yet its impact, albeit non negligible, can be controlled with proper tuning [15]. Therefore, we do not consider such metrics in this work. The two metrics are, hence, sufficient to compare the impact of strategies and configuration parameters on the scalability of the CN and the required investment to set-up the network, i.e., its capital expenses (CAPEX). More elaborate metrics can be used to assess the operating expenses of the network (OPEX). Some of them are discussed in Section 8.

## 5. Design Choices, Empirical Validation and Qualitative Analysis

To test and validate our tool we use Open Data from four areas in the Tuscany region, in Italy. There are three good reasons for this choice: first, topological data are available for these areas; second, some of the authors are natives of this region easing the access and interpretation of data that are often available in the local language; and third, the familiarity with the area helps to correctly interpret the results.

To facilitate the extension of this analysis to areas outside Italy with a consistent methodology, we use the already mentioned territorial characterization that is standardized by the OECD and divides territory in “Predominantly Urban” (PU), “Intermediate”, and “Predominantly Rural” [16]. The classification is based on the percentage of population living in urban / non urban areas, population density, percentage of countryside population, and the presence of large cities, and is intended to facilitate numeric comparisons between similar areas across different countries. In Italy, the OECD categories are further refined to make the classification more precise. A subset of urban areas, featuring prevalence of flat land, proximity to a city and advanced economy, is defined as “Rural with Intensive Agriculture” (RIA) and is basically a suburban area. Rural areas, on the other hand, are separated into “Rural Intermediate” (RI), which include hilly and some mountainous areas far from cities, and “Rural with Development Problems” (RDP), which include mostly mountainous areas that are generally far from cities and face development problems. Table 2 summarizes the main features of the four scenarios we consider, re-labeled urban, suburban, intermediate, and rural for the sake of easy reference.

If not otherwise stated, we apply the stopping criterion formulated in Section 3.4 with  $R_{thr} = 1$  Mbit/s and percentile=10, i.e., the network stops growing when more than 10% of the nodes can be granted less than 1 Mb/s guaranteed bandwidth. We stress that this is an estimate of the worst-case guaranteed bandwidth a user

acquires if all users are simultaneously active in the network. For most of the time, only a subset of users are simultaneously active in the CN and users will enjoy a much higher connection speed. ISPs generally use a ‘contention ratio’ or multiplexing factor of  $X:1$ , thus advertising speed that can be  $X$  times higher than the minimum guaranteed bandwidth per node. Typical contention ratios are in the order of 50:1, as recommended by the British telecom regulator authority in 2016<sup>8</sup>. Hence, a minimum guaranteed bandwidth of 1 Mb/s can be advertised as a 50 Mb/s connection, which is comparable to a typical FTTC connection.

### 5.1. Empirical validation

Before proceeding with large-scale analysis of potential CN growth we gathered data from one CN based in Florence, for which we were able to collect the positions of the nodes that form a network with 14 links. Since Florence is part of our dataset we validated our tool against these links; namely, we verified how many of those links our generator considers feasible. The simulator indeed verified the feasibility of 11 links and produced 3 false negatives. One reason for this small deviation was that the device-mounting pole in one of the CN nodes was much taller than the 4 m assumed in our simulator. Another cause of deviations is that real nodes are manually positioned on the building surface so that connectivity is optimized; there is no counterpart of such optimization in our simulator.

We conclude that our generator can estimate with adequate precision the macroscopic features of a CN, even if human intervention and careful node positioning may improve the feasibility of specific connections.

### 5.2. Qualitative analysis of one run: local strategy

In this section, we discuss indicative runs and we plot the tool metrics for a single execution of our generator, in order to get qualitative interpretation of the results. In Section 6 we present aggregate results over multiple simulation runs. For this purpose we generate a CN topology in an intermediate scenario area, using the local link selection strategy. In the selected run, the simulator was able to generate a CN made of 227 nodes, before the stop condition was triggered. The final CN graph has 907 (bi-directional) edges and the average CN node degree is roughly 8. Since the average number of devices per node is 3.67, we conclude that each device connects an average of 2.44 neighbors, or, in other words there are on average 2.44 links per device.

Figure 4 reports the Empirical Cumulative Distribution Function (ECDF) of the node’s degree (upper plot) and the minimum guaranteed bandwidth for all nodes (lower plot). Roughly 80% of the nodes have fewer than 12 neighbors (four devices with three links each), but there are a few outliers with up to 42 neighbors. The reason for this is that, although we limit the number of outgoing links per device to three, we do not limit the number of incoming links. Since the scenario area is geographically small, some nodes in the center of the area are connected to many neighbors. Note that all these links are not necessarily used to route traffic to the gateway, but they offer redundancy in the case of failures.

<sup>8</sup>The 2016 recommendation of the British Regulator considers 50:1 a good contention ratio for a standard broadband profile[17]

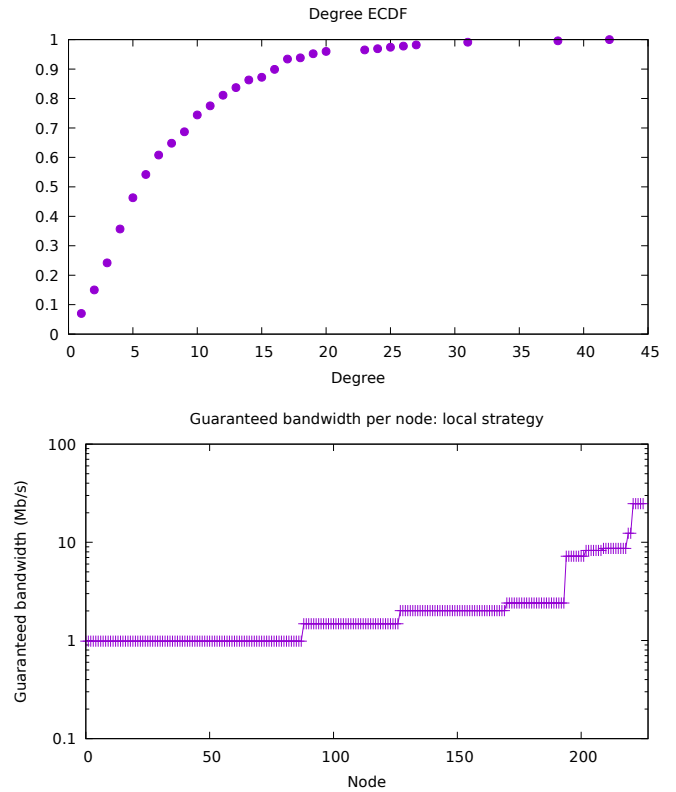


Figure 4: Intermediate area, 227 node CN. ECDF of the node’s degree (upper plot); minimum guaranteed bandwidth for all the nodes (lower plot).

The minimum bandwidth is quantized, as it is given by the bandwidth of the bottleneck link from any node to the gateway. The same link is a bottleneck for more than one node, so groups of nodes end up with the same minimum bandwidth. Note that there are 88 nodes ( $\approx 39\%$  of the total) with a minimum guaranteed bandwidth of 0.98 Mbit/s. This happens because before the last node was added to the network, one link with an estimated bandwidth of 87 Mbit/s was used by 87 nodes. When the last node was added, this pushed the bandwidth of all the 88 nodes using that link below 1 Mbit/s and terminate the CN growth process.

The upper plot in Fig. 5 reports the 10th, 50th and 90th percentile of the guaranteed bandwidth per node versus the number of nodes in the network. In line with intuition, as the network grows all percentiles decrease, meaning that the overall available bandwidth decreases with the number of nodes. The 90th and 50th percentile are very close, as it emerges also from Figure 4 which corresponds to the last point of the plot we are considering, in which the large majority of the nodes (194 over 226) get less than 3 Mbit/s. The lower plot of Fig. 5 reports the trend of the average cost per node in the network, together with the average number of devices per node. There is a fixed cost (200 €) per node, increased by the cost of devices as derived from on-line resellers of Ubiquiti devices. Prices range from 49 € to 200 €, but the difference in the performance of the devices emerges only on very long links. For links below a few kilometers, the mid-range devices perform as the high-end devices, so high-end devices are rarely used. For this reason the average cost of the nodes is directly correlated with the number of devices per node.

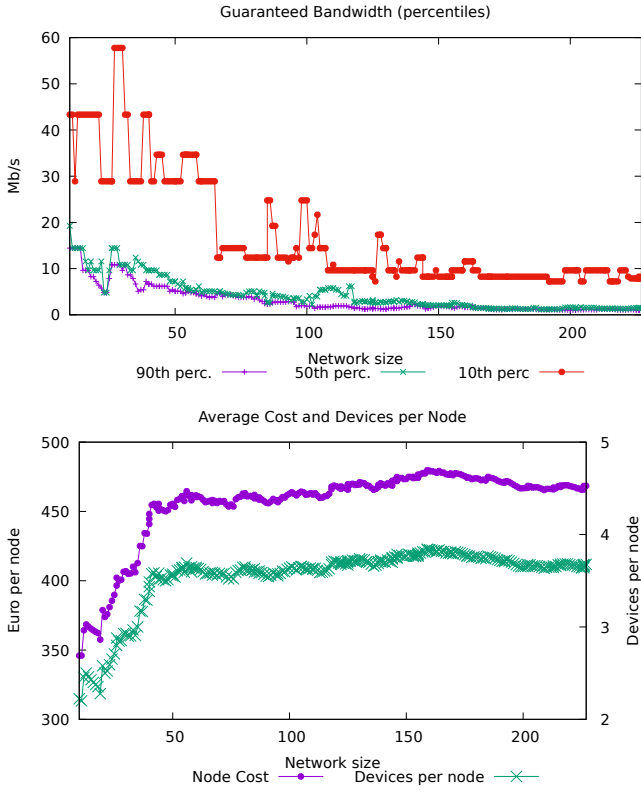


Figure 5: Intermediate area, 227 node CN. 10th, 50th and 90th percentiles of the guaranteed bandwidth (upper plot) and average cost and number of devices per node (lower plot) as a function of the network dimension (no. of nodes).

## 6. Quantitative Evaluation

Having set the metrics interpretation and validated the tool comparing its results with an existing network, we now perform a quantitative evaluation of the CN growth potential for the four area categories (urban, suburban, intermediate, rural) and both link selection strategies. As a free parameter to compare different “requirements” we use  $R_{thr}$  ranging from 1 to 5 Mbit/s; the stopping condition is 10% of the nodes below  $R_{thr}$ . Each point in the plots corresponds to the average of 10 runs and the error bars represent the standard deviation. In each run, we fix the gateway node in a different position and change the initial seed of the random generator that produces the sequence of requests.

### 6.1. Local link selection strategy

Figure 6 presents the three key metrics for the local link selection strategy for all four scenarios: the average number of nodes (upper plot), the average of the variable part of the cost per node (middle plot), and average number of devices per node (lower plot). The average size of the network is roughly proportional to  $\frac{1}{R_{thr}}$ , without significant differences among the four scenarios. Hence, what primarily influences the network size is the available bandwidth toward the gateway, and not the scenario. The average cost per and the number of devices per node (middle and lower plots), instead, clearly differentiate one scenario from another. Two factors dictate the number of devices per node: the choice of the neighbor when a new node is added and the terrain characteristics. We will discuss

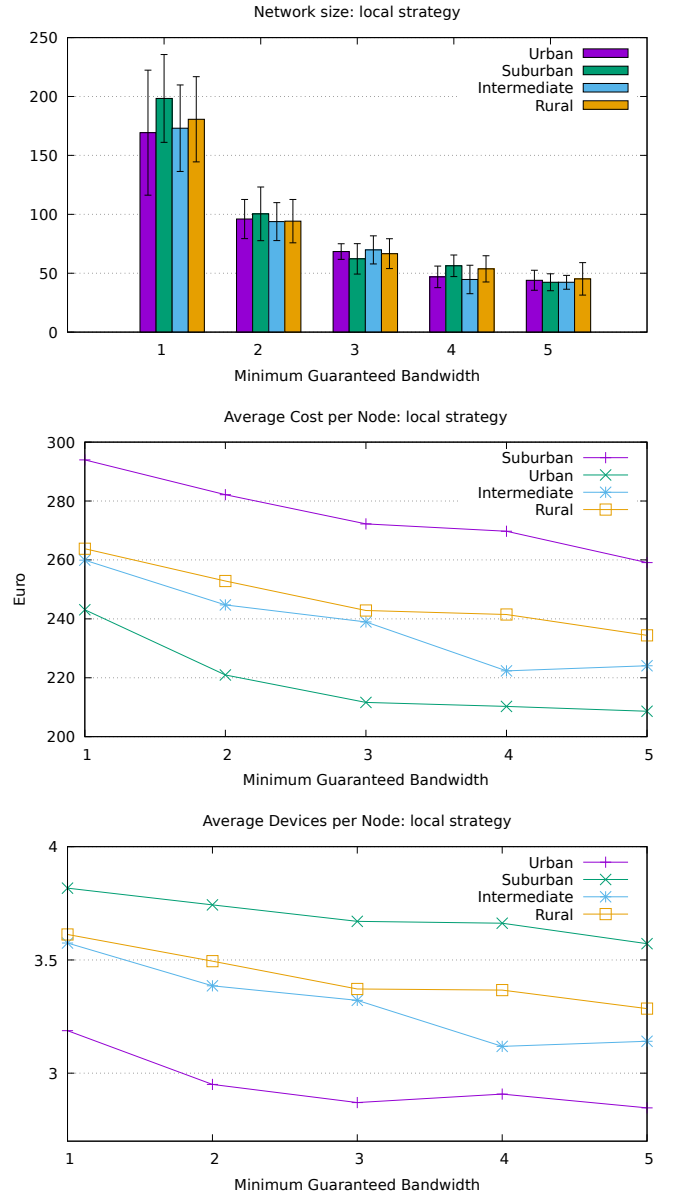


Figure 6: Local strategy: number of nodes in the network (upper plot), average of the variable part of the cost per node (middle plot), and average number of devices per node (lower plot); as a function of the minimum bandwidth  $R_{thr}$ .

the former in Section 6.2, while for the latter, it is intuitive to note that the building density plays a key role. When a new device is added, the antenna creates a cone in which futures node could be connected. The higher the building density, the higher the chances that a new node could be connected to an existing device. On the other hand, buildings are themselves radio propagation obstacles, which constrain the LoS between two network nodes.

Apparently, in the urban scenario the density of buildings suffices to reuse many of the existing devices, whereas the worst trade-off emerges for the suburban area. The middle plot of Fig. 6 shows that the impact of the scenario on the variable part of the node cost is pretty high (recall that we consider 200€ per node as a fixed cost). It would be extremely interesting to explore how the efficiency of the mesh network technology varies with the scenario and what

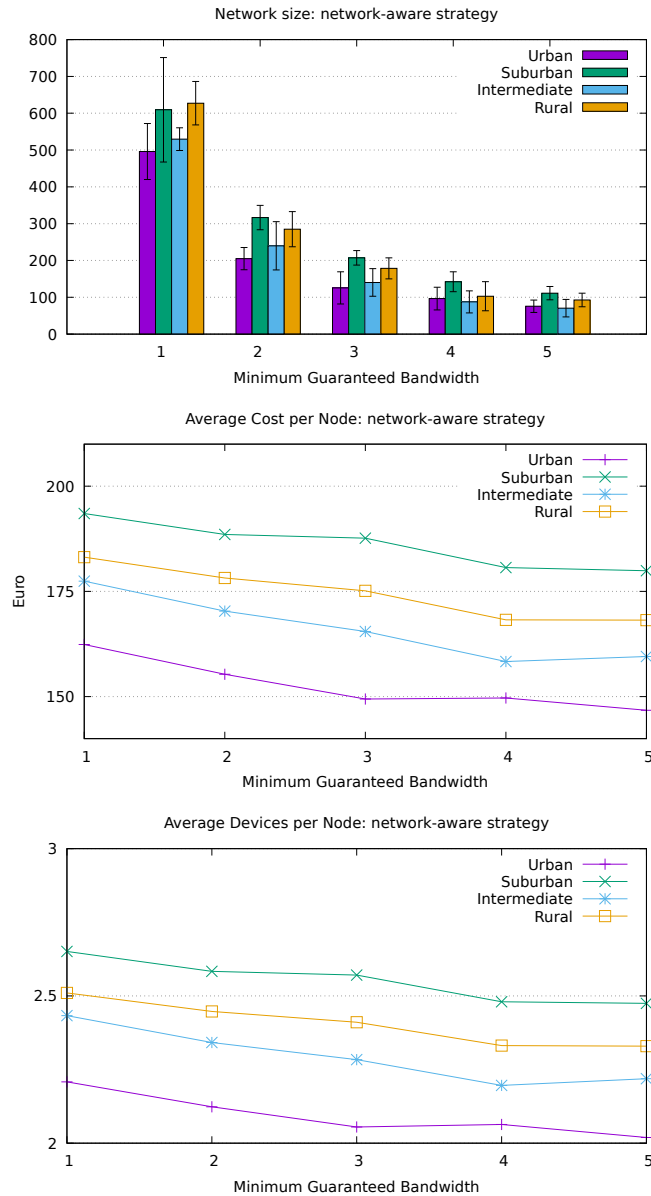


Figure 7: Network aware strategy: number of nodes in the network (upper plot), average of the variable part of the cost per node (middle plot), and average number of devices per node (lower plot); as a function of the minimum bandwidth  $R_{thr}$ .

are the design parameters we can modify (number of devices, type of devices, choice of neighbors) to improve it in each case, but this analysis goes beyond the scope of this contribution and is left for future work.

## 6.2. Network-aware link selection strategy

Figure 7 shows the same metrics for exactly the same areas selected as Fig. 6, but adopting the network aware link selection strategy. Compared the two the gain is considerable: the number of nodes is more than double all the scenarios for almost all guaranteed bandwidth values. Another positive effect is the considerably lower average cost per node. The local algorithm myopically tries to provide the highest capacity to each new link. This often results in the addition of new devices on existing CN nodes since a new

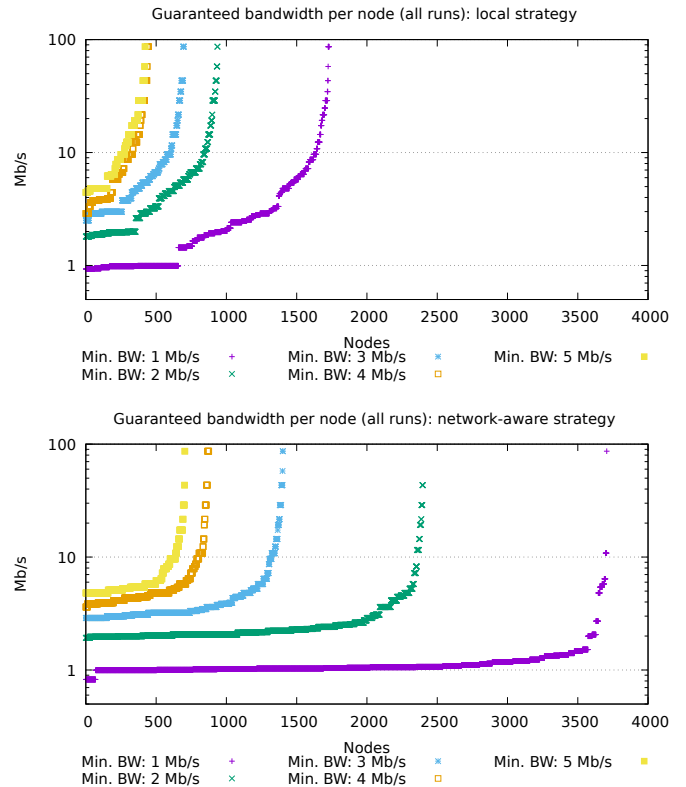


Figure 8: Guaranteed bandwidth to all nodes in all runs: intermediate scenario: local strategy (upper plot) vs. network-aware strategy (lower plot).

device always performs better than a shared device. However, the high-capacity links in the fringes of the network do not help much to increased the path capacity, whose bottleneck is normally is some central link towards the gateway. The network-aware algorithm, instead, has no specific incentive to add new devices unless this produces a network-wide improvement. Putting side by side the middle and lower plots of Fig. 7 and Fig. 6, show that both the average number of devices per node and the node cost decrease when the network-aware strategy is used.

Finally Fig. 8 generalize the data shown in Fig. 4 reporting the bandwidth that each node in every network generated for the intermediate scenario can expect to have at saturation. The other scenarios display similar behaviors and are not reported for the sake of brevity. Comparing the upper (local strategy) and lower (network-aware strategy) plots it is evident that this latter not only allows the CN to grow much larger doubling the number of nodes included, but also results in a more efficient use of resources, with the bandwidth at saturation distributed more evenly, which obviously imply that links and devices are used more efficiently as testified by the lower cost per node.

## 7. Related Work

Our work combines elements from two, originally distinct, operations: network planning and topology control. The first one is a longer-term centralized process that is carried out top-down and concerns primarily node placement in static networks. The latter is carried out over shorter time intervals and is more relevant to ad-hoc

networks dynamic networks. It involves the control of the transmit power of nodes in order to achieve certain performance objectives such as energy savings or resilience to node and link failures. Our focus is a combination of the two in bottom-up Community Networks.

The authors in [18] propose, analyze and optimize a cone-based distributed topology-control algorithm for minimizing the transmit power of ad-hoc network nodes, while preserving their connectivity. At the core of the algorithm lies the finding that, when focusing the nodes' transmissions within cone areas, there is some minimum power that can ensure their connectivity. Li and Hou in [19] are concerned with routing redundancy as a response to the increased risk of node failures or departures. Hence, they propose a fully localized algorithm that enforces  $k$ -vertex connectivity in the constructed topology while minimizing the maximum transmission power used in the network.

More relevant to our work is the thread on wireless mesh network planning, which has been primarily driven by the need to deploy wireless mesh networks in rural areas with limited or no network coverage. In [20], Chandra et al. consider the placement of Internet gateways (they call them Internet transit access points (ITAPs)) in a wireless neighborhood mesh network route traffic from residential nodes equipped with low-cost antennas to the Internet. They develop ITAP-placement algorithms that perform close-to-optimally over a number of scenarios addressing the neighborhood layout, end user demands, and the propagation environment.

A more holistic approach, including the selection of tower heights and antenna types, and aiming directly at minimizing the network infrastructure cost, is introduced in [21, 22]. Motivated by projects addressing the digital divide in rural India, the authors formulate the planning problem and then decompose it, exploiting dependencies between the different design variables and heuristics. The authors seek to optimize the selection of links to establish such that all nodes are connected and the resulting cost of antenna tower construction is minimized. [22] proposes a greedy algorithm that provides an  $O(\log n)$ -approximation.

Last, the wireless mesh network deployment costs are subject to a budget constraint in [23], where the objective is set to maximize the coverage of the users while ensuring that the network is resilient to node failures. The authors propose an approximation algorithm called greedy selection rounding (GSR), which persistently generates topologies with coverage at least 95% of the optimal at a cost that does not exceed by more than 15% their budget.

All these works assume a centralized, a-priori planning. In our case, the network grows bottom-up, in participatory manner, with users' locations determining the possible node and link additions. The CNO team can intervene in local choices to shape them in ways that facilitate their sustainable scaling, but cannot decide who participate in the CN.

The problem of network deployment optimization has been addressed in other areas such as Wireless Sensor Networks [24] since sensor nodes can be relocated, or relay deployment and optimization in cellular networks [25, 26], but the goals and constraints in these problems make them difficult to compare with the topic we address.

There is a rich scientific literature dealing with graph generators, since every research field that deals with graphs needs to design and test algorithms on realistic topologies. A good recent summary is the book from Newman [27].

Graph generators, however, produce graphs with some specific feature, while our problem is generating graphs with features that emerge from the local constraints of the networks deployment. Spatial networks are used to represent graphs in which nodes have geographic coordinates [28]. They enable a more realistic analysis of the trade-offs between distance and performance [29], but still they can not capture the specific features of communication network.

Finally, on the topology generators' front, a few attempts exist in the literature to build WMN planning software tools starting from the observation of real world mesh networks. Among them Cerdá-Alabern studied the topological features of the Guifi.net network [30] and derived a corresponding generator function, while Milic and Malek studied two networks of the Freifunk community and produced a geometric model that seems to be the most accurate model for mesh networks topology generation [31]. These models capture the macroscopic features of a network (like the degree distribution), but lack details to characterize their behavior in the real world, and do not offer a planning tool.

Other related fields are topology and economic analysis. Some researchers collected and analysed annotated topologies coming from existing networks. The data coming from the Guifi, ninux, and FunkFeuer networks have been analysed in the literature [3, 4]. The economic aspects of a mesh network, and its sustainability as a competitive communication infrastructure are analyzed in [2, 32, 33] and offer interesting expansion directions for the analysis tool we presented.

## 8. Conclusions and Directions for Further Work

Planning a bottom-up initiative like a Community Network is an oxymoron, yet even grassroots enterprises need a road map and a bit of design. With this paper we have proposed a tool that can be used to model and control the growth of CNs based on the local geographic constraints and additional economic parameters.

This work presents two key contributions. For the first time, the way Community Networks grow has been modeled as an (implicit) stochastic graphevolution with realistic constraints [34], also recognizing that the network evolution is first-order Markovian, i.e., the current topology and the new node that wants to join the networks are sufficient to describe its future evolution. Based on this conceptual model, we have been able to build a tool that simulates the evolution of a Community Network given the topography and the building database of the area, where the network is deployed. Moreover, we have modeled the standard way CNs evolve recognizing that this a local, egoistic and a bit myopic strategy that may hamper the future growth of the network. We have also proposed a network-aware strategy with acceptable computational complexity that is based on the global network benefit instead of local decisions.

Running the tool on urban, sub-urban and rural areas in Tuscany, Italy, where we know there is interest for founding CNs, we have shed light to the reasons these networks may fail, i.e., we were able to show that the network-aware strategy allows achieving networks that include more than twice the nodes of the local, myopic strategy with a cost per node significantly lower.

The tool we have presented, that is available open source, can be extended in several ways both as a planning tool and as a feasibility analysis tool.

The choice of next building to connect to the CN could be based on the distribution of population in the area under question, as this can be extracted from census data. European and US census campaigns report the number of citizens at granularities down to the city block in urban areas. Using open data we could then distribute people among the buildings in each block based on the buildings 3D volumes. Census data also hold information about the age, sex and education of people living in the block, and this data is correlated with the demand for connectivity.

The behaviour of more routing protocols in our model, beyond the simplified version of OLSR we have considered in this work, can be included in the tool, specifically for planning. Distance-Vector protocols such as Batman or BMX use Bellman-Ford to compute the shortest path and use different link and path metrics, and it would be interesting to compare the performance of different metrics and different protocols. In the long run, we plan to support also different communication standards, such as the mm wavelength communications, which are an integral part of 5G and need line-of-sight to communicate. Since wireless backhaul is an open challenge in 5G communications, and network reconfiguration is a hot topic, we believe that providing topologies that come from realistic constraints would be extremely useful for the scientific community.

## References

- [1] International Telecommunication Union, Trends in Telecommunication Reform 2008: Six Degrees of Sharing, Tech. Rep. D-PREF-TTR.10 (July 2009).
- [2] P. Micholia, M. Karaliopoulos, I. Koutsopoulos, L. Navarro, R. Baig, D. Boucas, M. Michalis, P. Antoniadis, Community Networks and Sustainability: a Survey of Perceptions, Practices, and Proposed Solutions, *IEEE Communications Surveys & Tutorials* 20 (4) (2018) 3581 – 3606.
- [3] D. Vega, L. Cerda-Alabern, L. Navarro, R. Meseguer, Topology patterns of a community network: Guifi.net, in: *IEEE Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Barcelona, Spain, 2012.
- [4] L. Maccari, R. Lo Cigno, A week in the life of three large Wireless Community Networks, *Ad Hoc Networks* 24, Part B (2015) 175–190.
- [5] L. Maccari, Detecting and Mitigating Points of Failure in Community Networks: a Graph-based Approach, *IEEE/ACM Transactions on Computational Social Systems* 9 (2019) 103–116.
- [6] S. Chen, K. Nahrstedt, Maxmin Fair Routing in Connection-Oriented Networks, in: *Euro-Parallel and Distributed Systems Conference (Euro-Par)*, Southampton, UK, 1998.
- [7] J. Kleinberg, Y. Rabani, E. Tardos, Fairness in Routing and Load Balancing, *Journal of Computer and System Sciences* 63 (1) (2001) 2–20.
- [8] K. Jain, J. Padhye, V. N. Padmanabhan, L. Qiu, Impact of Interference on Multi-hop Wireless Network Performance, in: *International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, CA, USA, 2003.
- [9] C. Barz, C. Fuchs, J. Kirchhoff, J. Niewiejska, H. Rogge, OLSRv2 for Community Networks: Using Directional Airtime Metric with external radios, *Computer Networks* 93 (2015) 324 – 341.
- [10] H. Rogge, E. Baccelli, Request for Comments 7797: Directional Airtime Metric Based on Packet Sequence Numbers for Optimized Link State Routing Version 2 (OLSRv2) (2016).
- [11] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. Lo Cigno, F. Dressler, How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help, *IEEE/ACM Transactions on Mobile Computing* 14 (7) (2015) 1411–1421.
- [12] A. Raniwala, K. Gopalan, T.-c. Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, *SIGMOBILE Mobile Computing and Communications Review* 8 (2) (2004) 50–65.
- [13] A. Neumann, E. López, L. Navarro, Evaluation of mesh routing protocols for wireless community networks, *Computer Networks* 94 (2015) 308–323.
- [14] A. Neumann, L. Navarro, L. Cerda-Alabern, Enabling individually entrusted routing security for open and decentralized community networks, *Ad Hoc Networks* 79 (2018) 20 – 42.
- [15] L. Maccari, R. Lo Cigno, Improving Routing Convergence with Centrality: Theory and Implementation of Pop-Routing, *IEEE/ACM Trans. on Networking* 26 (5) (2018) 2216–2229.
- [16] OECD (2011), Defining and Describing Regions, in: *OECD Regions at a Glance 2011*, OECD Publishing, Paris. doi:10.1787/reg\\_glance-2011-4-en.
- [17] UK Office of Communications, OFCOM, Achieving decent broadband connectivity for everyone. Technical advice to UK Government on broadband universal service (2016).  
URL [https://www.ofcom.org.uk/\\_\\_data/assets/pdf\\_file/0028/95581/final-report.pdf](https://www.ofcom.org.uk/__data/assets/pdf_file/0028/95581/final-report.pdf)
- [18] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, R. Wattenhofer, A cone-based distributed topology-control algorithm for wireless multi-hop networks, *IEEE/ACM Trans. on Networking* 13 (1) (2005) 147–159.
- [19] N. Li, J. C. Hou, Localized fault-tolerant topology control in wireless ad hoc networks, *IEEE/ACM Transactions on Parallel and Distributed Systems* 17 (4) (2006) 307–320.
- [20] R. Chandra, L. Qiu, K. Jain, M. Mahdian, Optimizing the placement of Internet TAPs in wireless neighborhood networks, in: *IEEE International Conference on Network Protocols (ICNP)*, Berlin, Germany, 2004.
- [21] S. Sen, B. Raman, Long distance wireless mesh network planning: Problem formulation and solution, in: *International conference on World Wide Web (WWW)*, Banff, Canada, 2007.
- [22] D. Panigrahi, P. Dutta, S. Jaiswal, K. V. M. Naidu, R. Rastogi, Minimum Cost Topology Construction for Rural Wireless Mesh Networks, in: *IEEE Conference on Computer Communications*, Phoenix, USA, 2008.
- [23] C. Chen, C. Chekuri, D. Klabjan, Topology Formation for Wireless Mesh Network Planning, in: *IEEE Conference on Computer Communications*, Rio De Janeiro, Brazil, 2009.
- [24] X. Wang, S. Wang, J. Ma, Dynamic deployment optimization in wireless sensor networks, in: *Intelligent Control and Automation*, Springer, 2006.
- [25] W. Guo, T. O’Farrell, Relay deployment in cellular networks: Planning and optimization, *IEEE Journal on Selected areas in Communications* 31 (8) (2013) 1597–1606.
- [26] P. Lin, J. Zhang, Y. Chen, Q. Zhang, Macro-femto heterogeneous network deployment and management: from business models to technical solutions, *IEEE Wireless Communications* 18 (3) (2011) 64–70.
- [27] N. Mark, *Networks: an introduction*, Oxford University Press New York, 2010.
- [28] M. Barthélemy, Spatial networks, *Physics Reports* 499 (1-3) (2011) 1–101.
- [29] R. Louf, P. Jensen, M. Barthelemy, Emergence of hierarchy in cost-driven growth of spatial networks, *Proceedings of the National Academy of Sciences (PNAS)* 110 (22) (2013) 8824–8829.
- [30] L. Cerda-Alabern, On the topology characterization of Guifi.net, in: *IEEE Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Barcelona, Spain, 2012.
- [31] B. Milic, M. Malek, NPART-node placement algorithm for realistic topologies in wireless multihop network simulation, in: *International Conference on simulation tools and techniques (Simutools)*, Rome, Italy, 2009.
- [32] A. Mazloumian, M. H. Manshaei, M. Félegyházi, J.-P. Hubaux, Optimal pricing strategy for wireless social community networks, in: *Proceedings of the 3rd Int. workshop on Economics of networked systems*, ACM, 2008, pp. 103–108.
- [33] R. Baig, L. Dalmau, R. Roca, L. Navarro, F. Freitag, A. Sathiaselan, Making community networks economically sustainable, the guifi.net experience, in: *Proceedings of the 2016 workshop on Global Access to the Internet for All*, ACM, 2016, pp. 31–36.
- [34] D. Carra, R. Lo Cigno, E. W. Biersack, Graph Based Analysis of Mesh Overlay Streaming Systems, *IEEE Journal on Sel. Areas in Comm. (JSAC)* 25 (9) (2007) 1667–1677.