Horizon 2020

Project title: Network Infrastructure as Commons

Monitoring Instruments for CNs

Deliverable number: D2.5

Version 1.0

Co-Funded by the Horizon 2020 programme of the European Union
Grant Number 688768

Project Acronym: netCommons
Project Full Title: Network Infrastructure as Commons.
Call: H2020-ICT-2015
Topic: ICT-10-2015
Type of Action: RIA
Grant Number: 688768
Project URL: http://netcommons.eu

Editor:	Leonardo Maccari, UniTn
Deliverable nature:	Report (R)
Dissemination level:	Public (PU)
Contractual Delivery Date:	December 31, 2016
Actual Delivery Date	February 24, 2017
Number of pages:	36 excluding covers
Keywords:	Community Networks, Monitoring tools, Centrality, Robustness, Social network analysis
Authors:	Leonardo Maccari – UniTn Renato Lo Cigno – UniTn
Peer review:	Merkouris Karaliopoulos - AUEB-RC Panayotis Antoniadis - NetHood

Executive Summary

This deliverable describes the work done to define a set of multi-layer graph metrics that Community Networks (CNs) can use to understand the degree of decentralization of both their communication and social networks. These metrics are a fundamental step toward a proper monitoring of CNs because they allow measuring several properties in network fragility and decentralization. This is needed because the evolution of a CN is generally unplanned, and there is no guarantee that, during its growth, the network evolves spontaneously in a robust and resilient network or it evolves toward a structure that contains single points of failure at one or more levels. These points of failure may reside in nodes that become topologically critical for the network itself, in infrastructure owners who own a set of nodes that give them too much potential influence, or in people in the social network of the community who become too much important in the discussion and in the organization of the community. In the worst option, the same person can represent a point of failure both in the communication and social network.

This deliverable formalises some of these concepts and introduces metrics that can detect the presence of such points of failure. The work in this deliverable is strongly based on two publications, one of which is currently available [1] and another is under preparation. To realize the publications, open source code has been released and is also partly documented in this deliverable.

The metrics and the methodology applied to the ninux network revealed that ninux is a theoretically decentralized network but in practice, at the time of our analysis had one single point of failure represented by a person that owned a number of critical nodes and was also very influential in the mailing list discussion. From this observation a heuristic algorithm to re-assign the property of some nodes was designed.

As part of the future agenda of Task 2.4 (T2.4), where this deliverable has been developed, we will propose the adoption of the monitoring software and of the proposed metrics directly in the web server used by the ninux community, in order to monitor the network development and avoid the future creation of more points of failure.

Contents

List of Figures

List of Tables

List of Acronyms

OLSR	Optimized Link State Routing Protocol
CN	Community Network
ISP	Internet Service Provider
CDN	Content Delivery Network
QoS	Quality of Service
IX	Internet Exchange
DSLAM	Digital Subscriber Line Access Multiplexer
WP1	Work Package 1
T2.4	Task 2.4

1. Introduction

The fundamental difference of a CN with a traditional Internet Service Provider (ISP) network is its decentralization, both at the technical and at the governance layer. This deliverable documents the experiments done on existing community networks to monitor the level of decentralization of the communication infrastructure and the social network, to understand to what extent those CNs can be considered decentralized networks.

Before getting deeper in this analysis this chapter gives an introduction on what centralized and decentralized means, what happens when a decentralized network grows, and the consequences of centralization and decentralization for the network users. It will review the basics of Internet access services, the way ISP networks work and the differences with a CN.

1.1. Network Decentralization

Let us abstract a network topology as a graph $G(V, E)$ made of a set of V nodes and a set of E edges. There is not a precise working definition that can be used to distinguish between a centralized/decentralized/distributed network, but even in the classical literature their tentative definition is related to the level of redundancy [2] that the network offers, and thus its resistance to failures (robustness). The level of decentralization then influences other aspects, such as the cost, the way the network is managed and its efficiency.

1.1.1. Decentralization and Robustness

In a network graph, one way to define its redundancy is the ratio between the number of nodes and the number of edges [2]. It is intuitive that a star network, in which all the nodes are connected with a link to the same hub, is a network that is not redundant: every time a link is broken, at least a node is isolated from the rest of the network. In the worst case in which the hub fails the network does not exist anymore. The concept or redundancy is thus meaningful if related to the robustness of the network to failures, the more the network is redundant, the more failures it can sustain. Since the goal of a network is to connect the nodes one to the other, resistance to failures is measured in the number of couples of nodes that can still communicate one with the other once a failure happens. Again it is intuitive that if in a network made of a set of V nodes and E edges the failure of one node makes it impossible for any other couple of nodes to communicate, then the network can not be called "robust". At the opposite extreme we find what is a called a "full-mesh" topology, that is a network in which every node is directly connected with an edge to all the other nodes. In this case, the failure of a single edge does not prevent any couple of nodes to communicate, there is a path made of one or two hops that connects any couple of nodes even after the failure of one edge. Moreover, the failure of a node does not prevent any other node in the network to communicate between each other. Such a network is intuitively more robust than the star network described before.

A star network is a network with a strong centralization of resources, and little robustness, a full-mesh is a network with a strong decentralization of resources, and high robustness. Thus, a way to define the level of decentralization of a network is to test its resistance to failures. This can be done a-posteriori once the network graph is already present, and it requires simulating the failure of network nodes (or edges) and count the number of couples that can still communicate in the network graph. Note that robustness must be interpreted not only as resistance to technical failures, but also as the potential resistance to behaviours or misbehaviours. In the star topology, the manager of the hub can control all the traffic in the network, he/she can spy on it, filter it, slow it down. If the star network connects different stakeholders that govern the network, the manager is also in a key position to settle the rules with the others, because his negotiating power is the largest one. So, while decentralization deals mostly with network properties, it has direct implication on network governance too.

1.1.2. Cost, Management and Efficiency

In terms of cost, the advantage of a star network (as an instance of a tree network) is that it can be realized with a minimal number of edges. To connect V nodes, exactly $||V|| - 1$ edges are needed (where the $|| \cdot ||$ operator refers to the size of a set), while the full mesh network uses the maximal number of edges ($||V|| - 1$ per each of the nodes in V). Since the cost of a network strongly depends on the number of edges, the edge-to-node ratio largely impacts the overall cost of the network.

Also in terms of network management a star network is easier to manage than a full mesh network. In the star network the whole network intelligence is concentrated in the hub. All the other nodes do not need to know anything about the rest of the topology, since the only configuration they need is the use of the hub as the default gateway necessary to reach the other nodes. The hub will need a routing table, which is a static table that assigns a physical port to the address of the node that it is connected to that port. Under a management point of view, this configuration is the easiest to maintain. The majority of the resources and the efforts are concentrated on the hub in terms of hardware, software, and in the needed personnel that technically manages the node.

In a full-mesh network instead, all the nodes are sitting on the same level of relevance, and thus, they include all the functionalities of the hub in the star configuration. As an explicative example, consider that every node in the full mesh must be equipped with a router with a number of input ports that scales linearly with the number of nodes in the network. When the number of hosts overcomes the number of ports in the router all the routers need to be replaced with larger ones.

Finally, the efficient usage of the resources is another key factor to consider. In general, the reason why a person needs access to a network is for his/her interest in the services that the network gives access to. These services are normally not uniformly distributed among the network nodes, so there are nodes that are more “interesting” than others. It makes sense to invest in connections to the interesting nodes and leave fewer resources for the connections to the other nodes. Conversely, it makes sense to place important services on network nodes that are reached by fast connections. As a consequence a loop is introduced: the more the services are centralized, the more the network capacity is increased around the important nodes, the more the important nodes become suitable to host more services. A star network embeds a natural hierarchy, so the place where to install services is the hub. In a mesh network services can be placed anywhere, which is an advantage, but it also means that the link between two nodes with no services will be underutilized, so some network resources will be wasted. Note that this point is particularly important because it introduces a feedback between the physical structure of the network and the use of the network made by its users, which is one of the focuses of this research project.

So far only toy-examples of network topologies have been discussed, star networks nd full mesh networks. There is no doubt that none of them can scale to serve up to thousands or millions of users. Real networks are a mixture of star, mesh and full mesh topologies that create multi-hop communication networks. In real networks the differences blur and it is not that easy anymore to understand if a network is centralized/decentralized/distributed.

The goal of Task 2.4 is to describe a methodology and publish the open source software needed to estimate the robustness of a CN. Since a CN is both a communication and a social network the methodology must take into account and relate both these layers.

Before describing the methodology the next step is to go deeper in the way real networks work and their features.

1.2. Access Networks

Since a CN is a communication network, it must be compared to the other communication infrastructure we know. If we look at the Internet we know that it is made of the interconnection of separate networks run by ISPs, whose goal is to give network access to users or other ISPs. We will now give an overview of its basics and then concentrate on access ISPs, since today CNs are mostly imagined as a replacement for them (but their role can actually be much more than that [3]).

1.2.1. Internet infrastructure, Layers, Tiers, Peering

Describing the Internet is problematic because there is no global planning of its function, no single owner that can design its development and thus no single point of view on its internals. A classical description of the Internet splits the ISPs into three different tiers. At the bottom layer, at tier-3, we have access ISPs that sell their Internet connection to the users. These are the ISPs that the households or the small companies sign contracts with. An access ISP can be of any size, from local ones with hundreds of customers to large ones with millions of customers. An access ISP buys the access to the rest of the Internet from another provider that sells Internet transit. These providers operate at a regional level and are called tier-2 providers. They in turn buy Internet transit from tier-1 providers who operate at global layer. Buying transit is not the only way ISPs interact, they can also settle peering agreements. A peering agreement is a business relationship between two providers that agree to freely exchange traffic one with the other. Tier-1 ISPs are all connected via peering agreements, so tier-1 ISPs do not by transit from other ISPs.

This picture was largely modified lately. Content Delivery Network (CDN) “flattened” the Internet architecture and peering was reduced by operators [4], but it gives a view of how much the Internet infrastructure developed as a layered infrastructure.

1.2.2. Access ISP networks

A traditional ISP network is layered, hierarchical and centrally governed. The reasons for this design are related with the consideration we did so far. The **cost** of the infrastructure is one of them.

Networks today mostly rely on wired connections, and the cost to deploy a wired connection is dominated by the non-technological expenses (such as roadworks), which represent between 70% and 80% of the total cost [5]. For this reason, edge-to-node ratio tends to be close to the minimum. Networks must have some degree of redundancy to tolerate failures but redundancy is normally applied to the core of the network, and far from end-users. A tree topology is the most cost-efficient topology so the closer to the user, the more the network has a tree-like topology. A tree-like topology implicitly introduces a layering in the network, from the leaf nodes to the root node. Leaf nodes are the end-users and the root is generally the border router that connects the ISP network to the Internet, via an Internet Exchange (IX). This is of course a very simplified view of the complexity of an ISP network, but serves for the purpose of comparison with CNs. More details on real network deployments can be found in the literature [6, 7, 8].

Having a layered network also automatically introduces a hierarchy in the network. The technology used at each layer of the network is different from the technology used at the layers below it. Home broadband connections are generally reached with copper cables and aggregated in Digital Subscriber Line Access Multiplexers (DSLAMs), which are connected via gigabit Ethernet of Optic Fibers links to the transport network of the ISP up to some IP gateway connected to an IX. Each technology needs different expertise to be managed and is in control of specialized operators. Having layers of different technology helps the **network management** based on a “divide et impera” strategy, and addressing problems with different technical operators. This is reflected, for instance, in the way technical customer support is organized. An ISP has different tiers of operators that can act only on some tier in the network, and when something can not be solved at their layer, the support request escalates to a higher tier.

An ISP network is dimensioned to sustain a certain number of users but it actually can not carry all the traffic needed by those users: ISPs oversell their resources. Overselling the network capacity is typical of packet-switched networks, imagine an ISP that buys transit capacity from another ISP of 1Gb/s, and resells Internet access to his customers up to 100Mb/s downlink. Intuitively the ISP can not have more than 10 customers, or else they could saturate the network when they all concurrently download traffic. In reality, Internet traffic is bursty, so the majority of the people use a fixed Internet connection only for a short time during the day, so the probability that all the ten customers of our example ISP decide to connect together are low. Using this assumption, the ISP can resell the network resources to a number of users that would theoretically need more capacity than the one available at the ISP. The term “contention ratio” is used to express how much users can be compressed into a pipe that would be sufficient to serve only one user. Contention ratios of 1:10/20/50 are used by providers to increase the efficient usage of their resources. Overselling network resources increases the revenues, but also helps in using the network more efficiently. If the network would be dimensioned to sustain traffic in worst case scenario (every user using all the available uplink and downlink bandwidth), for the large majority of the time the network would be underutilized. Overselling resources makes it possible to have a core network whose throughput always lies close to the maximum, so its cost is repaid.

If the network is technically under dimensioned the ISP must be very careful to avoid congestion as much as possible. An access ISP network generally has one point of ingress and egress to the Internet, and a number of leaf nodes that are receivers and generators of traffic. Access ISPs buy transit bandwidth and settle peering agreements with other ISPs, so they know what is the maximum ingress traffic rate that their network must sustain. The transport network of the ISP must be dimensioned in order to sustain this traffic and deliver it to the end-users. Once a data packet arrives at the border of the network and must be delivered to an user, ISP has to ensure that on the last link (the so-called last mile) there is enough bandwidth to deliver the packet. When a packet arrives at the last mile it has already spent resources in the core network which would be wasted if the packet is not delivered. Thus, the ISP has an interest in ensuring that the downlink bandwidth to the final user is high enough to sustain all the traffic that is delivered up to the last mile. Leaf nodes (end-users) can also generate uplink traffic that will be egress traffic from the ISP network. While the ingress traffic is capped by the commercial agreements the ISP signs with other ISPs, the amount of internally generated traffic is in principle unpredictable, it depends on the number of users and on their activities. For this reason ISPs have an interest in reducing this traffic to the minimum, or else, uplink traffic could congest the ISP core network. This is why ISP tend to sell asymmetric bandwidth to their users, because a high downlink is easier to manage than a high uplink. Having asymmetric links guarantees that congestion can not happen bottom-up (from leaf nodes to father nodes) but only top-down, and thus it is easier to control from the network manager. Aside this reason, there are legacy reasons that come from the technologies used for the last mile, which were designed from the beginning with asymmetric performance, as a consequence of the client-server model that was the most common in the 90s.

As a consequence, users buy subscriptions to the Internet without knowing their real performance. Most of the fixed broadband Internet access contracts specify that the connectivity can achieve up to a certain speed that is in the order of tens of Mb/s , while they do specify a guaranteed speed that is in the order of tens of Kb/s. In practice, the lower bound of the available bandwidth is required to guarantee that a service is formally available but is too low to enable any real use.

Note that a tree-like network is easier to design than a mesh network. Once the number of leaf nodes is known in a certain area (or there is a reasonable forecast of it) the link starting on their father node is dimensioned to carry enough traffic for that number of users, and the same procedure is repeated at each tier. In a mesh network the path that goes from a gateway node to some node A is not easily predictable, thus it is not easy to estimate how many traffic streams will be using a certain link in average.

Finally, an ISP is generally run as a for-profit activity, and thus, the **governance** of the ISP is in the hands of its stakeholders. Infrastructure planning is part of the business planning of the company: The number of houses that the network can reach, the quality of the connection, the capacity allocated to each user is part of a design strategy that is internal to the ISP and on which the users have no control at all.

1.2.3. Consequences of Centralization for the Users

As a consequence of this approach users are excluded from any form of control on networks. **A layered network is a network that hides the complexity of higher layers to the lower layers, it is an opaque network by design**. Broadband users can normally control the network devices placed in their house, they can check the availability of physical connection to the next hop, but have no power in understanding what is the performance of the network beyond their home router. The network topology is not known so there is no way to understand at what hop starting from home to the ISP gateway (or beyond) a network problem occurs. Network troubleshooting software can be used, but without a big picture of the network topology they are of little use. Similarly, the network design specifications are not known to the users, who have no chances of accessing them. Users do not know the contention ratio and, more generally, the resources that are reserved to each of them. For the same reason users do not know what are the Quality of Service (QoS) policies applied by the ISP. The traffic from and to the Internet is managed by the ISP routers that can apply traffic shaping in order to prioritize some traffic. Users have no way to understand if a certain behaviour is due to a condition of congestion of the ISP network, that is overselling beyond the limit of usability, or it is due to a deliberate choice of an ISP to slow down certain traffic (this happens with P2P traffic, for instance [9, 10, 11]).

Asymmetric bandwidth is another element that precludes user of independence. Its initial justification was driven by the dominant client-server model, but today it favours the growth of centralized services against decentralized ones. Decentralized services can not compete with cloud-based ones if the cloud-based ones exploit a downlink bandwidth that is 10 or 20 times higher than uplink bandwidth. Asymmetric bandwidth facilitates the use of centralized services, which in turn are known to be a threat to user privacy [12].

Finally, when the user does not have any control on the business decision of ISPs, digital divide grows, because ISPs have no interest in reaching areas that do not guarantee a minimum level of revenues.

1.3. Decentralized Access Networks: CNs

CNs are mesh networks, and the majority of the CNs start as “wireless” CNs. This is not by chance, because the cost of deploying a wireless link is orders of magnitude lower than the cost of deploying an equivalent wired link. Theoretically, this technological change alone introduces some key differences in the network topology. These differences ignite a whole set of consequences that leads to the empowerment of the users.

First and most important, the network density does not need to be as low as with a wired network. Mesh networks allow a level of redundancy that a tree network does not have, starting from the periphery of the network. In our previous studies we observed that in the three networks we studied the number of leaf nodes is below 42% of the whole nodes [13]. This means that theoretically the robustness of users connections is increased compared to a standard home broadband connection, since the failure of one link that connects a node may not prevent the node to be connected to the network via another link.

Higher density makes it possible to build network topologies that are more flat and less hierarchical than traditional networks, and CNs have an interest in keeping the network architecture as flat as possible. This is due to two reasons, the first is that there is an objective difficulty in planning a network that evolves spontaneously. Nodes are added when some community member joins the network and the network evolves following the so-called “network effect”: the more people join the network, the higher the network coverage, the higher the chances that more people will be able to join. After a certain size that depends on a number of factors the network must be split in separate subnets, which implies that a “backbone” network will be needed to connect the various subnets. Still, the larger is each independent subnet, the more agile becomes the network management. If the network evolves in an unplanned way it is hard to layer it and define a hierarchy and a technological separation of layers. The second reason for keeping the network flat is that this way, every person in the community can participate to its maintenance without having to create a hierarchy of roles. If technical maintenance is needed to a certain node, anybody in the community can possibly have the skills needed to apply it: network management is not layered. Of course not all the community members are able to repair or reprogram a node, but there is an effort in sharing the knowledge necessary to let as many people as possible to manage their own nodes.

Another key element is that in a flat network there can be more than one gateway to the Internet, and every user can be directed to a gateway or another depending on his/her own configuration or even on the state of the network at that specific moment. This normally happens in a CN in which many people share their home connection (or they collectively acquire one or more than one broadband connection) with the other people in the CN. Under these conditions it is hard to define a traffic matrix and perform efficient network planning.

These technical features that influence the network topology render several of the social consequences we introduced for ISPs less likely in a CN. For instance, there is no interest in using asymmetric bandwidth. The general idea is that the traffic flows from the gateway to the leaf nodes that is generally true for an ISP instead fades in CNs. Every node is itself a router, there can be more than one gateway and internal services can be accessed without the need to reach a gateway. Each of the links that a node has can be used in both ways and there is no reason to impose different download and upload speed. Similarly, it is hard to make network planning in a mesh network, because the path from the source to the destination can not be known in advance. Mesh routing protocols should be able to distribute the traffic from the most loaded to the less loaded network areas, but this is not an easy process to forecast. Thus, the idea that there is a contention ratio in the network core is not easy to apply to a mesh network, the network resources are generally over provisioned, instead of under provisioned.

The fact that there is not a global view of the desired network coverage from the beginning of its deployment means that there is a implicit assurance that no one can be left out of the network for reasons that are not related to technical limitations. Every decision is taken by the community and can be discussed inside the community, there is no business plan to be satisfied that removes some choices from the possible ones.

Of course there are many drawbacks, and the one that represents the highest challenge for the future of CN is that it is still unclear how much a flat architecture can scale, both in terms of network capacity and in terms of community management. In this task we analysed community networks whose size is in the order of hundreds of nodes. Each node serves a family, a small business or an association, so probably thousands of people daily. We still do not know how much a flat architecture can grow maintaining its original asset.

1.4. Scaling up the Communications and the Social Network

Indeed, all the described features are true at a high-level, but the implementation of a CN may in the end be different than expected. While there is a feeling often reported in the CN narrative [14] that the technical features of a CN make mesh networks more participatory and democratic by design, the goal of this task, and of this deliverable, is to understand if this is the case, and if the joint evolution of the communication and the social network confirm this, or they show a different reality.

If we abandon the CN narrative and we look at the results in the literature, there are reasons to doubt that distributing a system can really improve its P2P nature and its participatory democracy. We review two important themes, from two different fields.

1.4.1. Scale-free Networks

In the past two decades, after the works of Barabási et al. [15] the study of networks has received a lot of attention and based on the so-called “Network Science” discipline [16]. The observation that started this trend is that many different networks share one feature, they are scale-free. Being scale-free means that the distribution of the degree of the network nodes follows a power-law, and thus, the scale at which the network is analysed does not change the way it macroscopically looks. Another consequence of this property is that a very small fraction of the nodes will have a very large number of neighbors, while a very large fraction of the nodes will have a small number of neighbors.

This observation reveals that in many networks there is a small number of very well connected nodes that represent a hub of the network and a majority of other nodes that are structurally irrelevant for the life of the network. In other words, even if there is no centralized planning it seems that for some unclear reason networks of different kinds tend to build a natural hierarchy. A direct consequence of this is that networks tend to be structurally robust to random failures: given that only a small portion of the nodes is critical to keep the network connected, if failures are randomly distributed among all nodes the probability that a critical node fails is low. On the other hand, they are fragile to targeted attacks: if an attacker can choose the node to tear down, he will pick one of the most important nodes and will have a strong effect on connectivity [17].

The emergence of scale-free topologies inspired a large body of research because it introduced some "universal" features of networks that are apparently uncorrelated, from biologic networks to the pattern of cooperation in social networks. It thus raised the question if all the networks, even the ones that were not planned to be hierarchical, for some reason tend to build a hierarchical infrastructure. This of course has an impact on the way these networks can be managed, and on the power that the owners of parts of these network can exercise on the network users [18].

It must be said, tough, that the works of Barabási and other scholars studying power-laws have been also criticised under several points of view [19]. In some cases the data-set used has been shown to be partial, and statistically the emergence of power-law distributions is not surprising under certain conditions, and networks probably do not represent an exception. While it is true that in the general case the scale-free features of networks received more emphasis that they probably deserved, the powerful simplification (or over-simplification some would say) of the scale-free narrative was able to break the boundaries of computer science and to attract the interest of different disciplines. In other words, it has the merit of creating a new interdisciplinary interest in networks. Without entering into details, it is interesting to understand if the scale-free property of networks apply also to CNs.

1.4.2. Michels' Iron Law of Oligarchy

Robert Michels in 1911 described the so-called "Iron law of oligarchy", referring to political organizations of the time. This social theory basically says that any organization that grows large and complex enough, independently from its initial efforts to keep a democratic and participatory approach will degenerate into an oligarchy [20, 21].

For sure this is not the right context for a deep analysis of this theory, but recent studies outlined that these patterns can be detected also in the digital production world. In particular a work from Shaw et al. [22] outlined that the degeneration to oligarchy can be found in some Peer Production platforms, such as Wikis. A Peer Production platform is an on-line platform that allows people to interact in order to create some shared good, material or immaterial [23]. Peer production is a lively research field which is flourishing due to the remarkable success of some peer production platforms, including Wikipedia or some Free Software movements. Researchers are today trying to understand if those decentralized, spontaneous and participatory movements are able to create new social value with a peer-to-peer approach. The work of Shaw actually points to the fact that some of the features identified by Michels can be observed also in some peer production platforms so that these peer-to-peer platforms can degenerate into factual oligarchies. It is again interesting for the study of CNs to understand if, under a governance point of view also CNs tend to abide to the "Iron Law" or follow some intrinsically different model.

2. Monitoring Decentralization in CNs

The goal of the first 10 months of T2.4 was to contribute to the research on CN analysing the data available from three community networks: the FunkFeuer network in Wien and Graz, and the ninux.org network in Rome (abbreviated respectively as FFWien, FFGraz, ninux). The aim of the analysis is to contribute to answer two questions:

- Is the evolution of the network graph different compared to other communication networks, such as scale-free networks?
- Given that the goal of the community is to build a distributed network with a de-centralized management, is the result close to the expectations of the community?

To answer this questions we collected data from the three networks, we realized open source software to analyse it and we described the most useful multi-layer metrics to understand the robustness of the networks, which we observed, is the measurable counterpart of decentralization.

2.1. The data-set

The three networks use Optimized Link State Routing Protocol (OLSR), a link-state routing protocol that makes it possible for each node to be aware of the whole network topology. The communities publish the network topology dumped by the OLSRd daemon, that can be used to analyse the network evolution. The topology recorded by the routing daemon can be misleading: in some cases a number of devices placed in the same physical location are attached to a wired switch and each of them runs a separate instance of the routing protocol. For OLSR they are different nodes but, in practice, they are not. To merge these cliques, another source of information is needed. More details about the networks and the merging technique can be found in the published source code, which will be briefly illustrated in the appendix and in previously published works [24][13].

The FunkFeuer networks publish a long history of dumps, while for Ninux only the current state is available, plus data collected in a week-long monitoring realized in 2014 [13]. Table 2.1 reports a summary of the available data, in the following of this report the time based evolution of the network always refers to the FreiFunk networks, instead when the analysis is done on a single snapshot, the sample with the largest number of nodes for each network in considered.

For the ninux network, two other sources of information were accessed. The first is a database containing the mapping between the physical node and an email of a person that owns it, the second is the archive of the mailing lists of the ninux community of Rome for the year 2014.

	FFWien	FFGraz	ninux
maximum recorded nodes	235	126	140
maximum recorded links	450	181	158
time series available	yes	yes	limited
first dump	2013-07-27	2007-03-31	2014-1-14
last dump	2014-02-15	2016-02-21	2014-1-20
dump interval	weekly	monthly	every 5 min
node ownership	no	no	yes
mailing list	no	no	yes

Table 2.1: The summary of the available data

2.2. Related Works

CNs have been the subject of a series of research works in the past years that had the goal of analysing their topological features [25][24][13][26] their routing solutions [27][28] and their social and management aspects [29][30]. This work performs a different analysis based on two original elements, the first is the analysis of the time-evolution of the networks, which helps understanding what was, and potentially what will be the evolution of the network. The second is the mixed social and technological analysis aimed at identifying single points of failure in the techno-social organization of the network. The only work in the literature that deals with community networks and uses a similar approach is from Vega et. al. and analyses the Guifi.net community [31]. Guifi is probably the largest community network in the world, and the analysis of the mailing lists and people interactions done at that scale can give general results. Our work instead was focused on a smaller network, but started from previous observations and interviews that we realized in that environment. In a more controlled environment and with direct contacts with the network participants it is easier to draw solid conclusions on the techno-social dynamics of the CN.

2.3. The Network Graphs and their evolution

Figure 2.1 fig. 2.2 and fig. 2.3 show the relative frequency of the degree distribution for the three networks, and the best-fit with a power-law function. A power-law degree distribution is normally observable in the central part of a distribution or in the right tail and requires roughly a couple of orders of magnitudes on both axes. In this case the size of the networks (hundreds of nodes, and maximum degree that ranges from 11 to 29) makes it statistically hard to identify a trend.

An alternative approach is to investigate if the network evolution follows a preferential attachment model, which would lead to a more evident scale-free behaviour with the growth of the network. The preferential attachment model describes the way in which new nodes are added to the network, and the entry points they connect to. In such model the rate $\Pi(k)$ with which a node with k links acquires new links is a monotonically increasing function of k. Following a preferential attachment model is not a necessary condition to have a scale-free network, however since it has been shown that it is at the base of several different kinds of scale-free networks (the Internet graph for instance, has been shown to have $\Pi(k) \propto k$ so that the probability of acquiring now links is proportional to the current number of links of a node [32]) measuring the relationship between $\Pi(k)$ and k can give insights on the future evolution of the network. This behaviour is easier to test on this data-set because the total

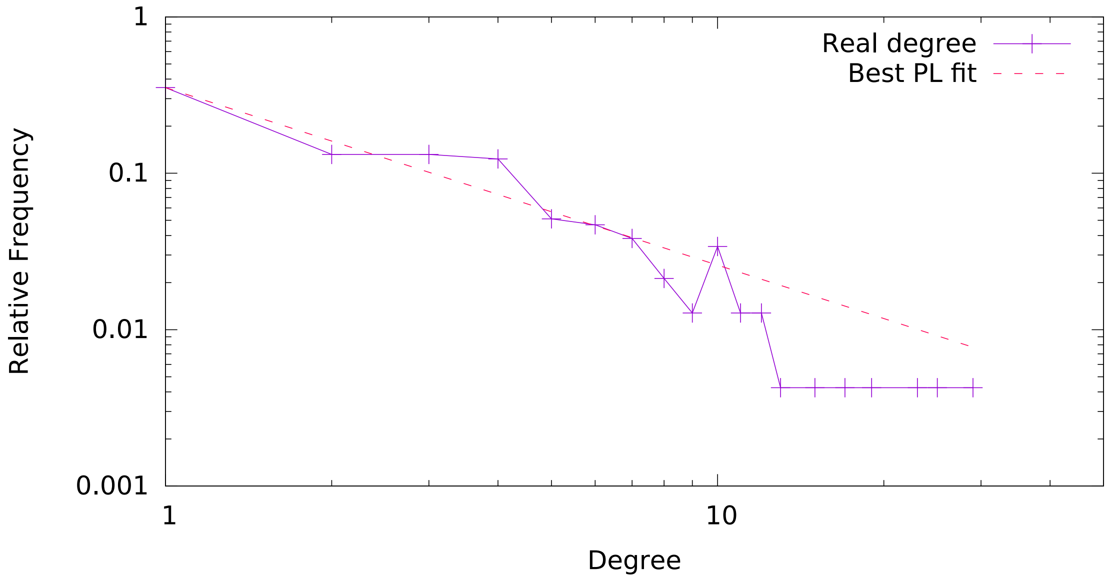


Figure 2.1: The degree distribution for FFWien, and the best power-law fit $x^{-\alpha}$, $\alpha = 1.13$.

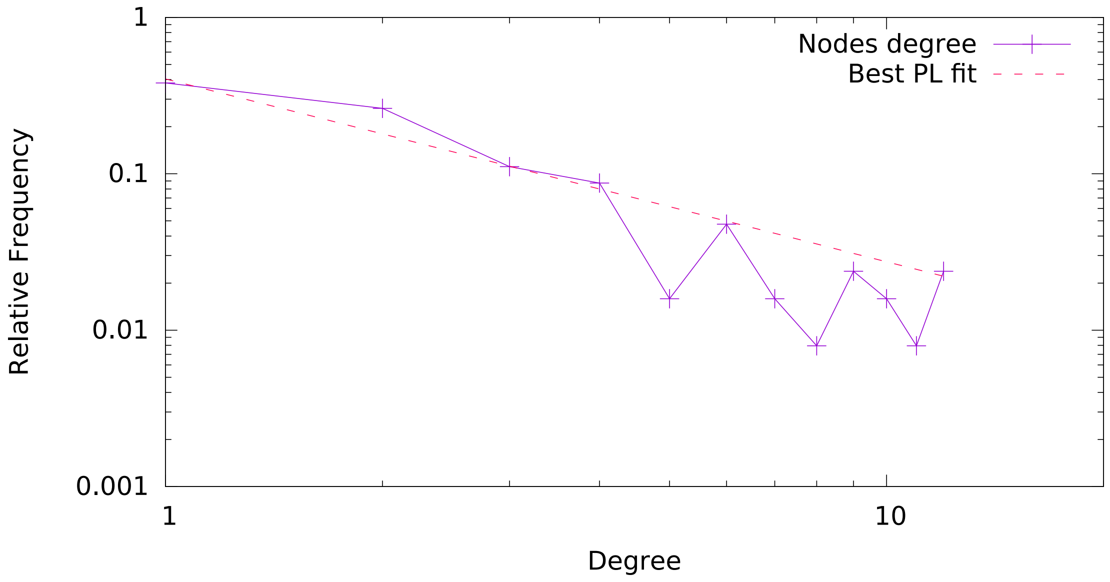


Figure 2.2: The degree distribution for FFGraz, and the best power-law fit $x^{-\alpha}$, $\alpha = 1.16$.

number of new nodes that joined the network during the observed period is much higher than the number of nodes at the end of the interval, since many nodes join the network for a limited period of time. To test the hypothesis of the preferential attachment model, for each year of the available data, for every new node added to the network the degree of the entry node was recorded and collected in a histogram that approximates $\Pi(k)$ (each node is counted only once at its first entry). To smooth

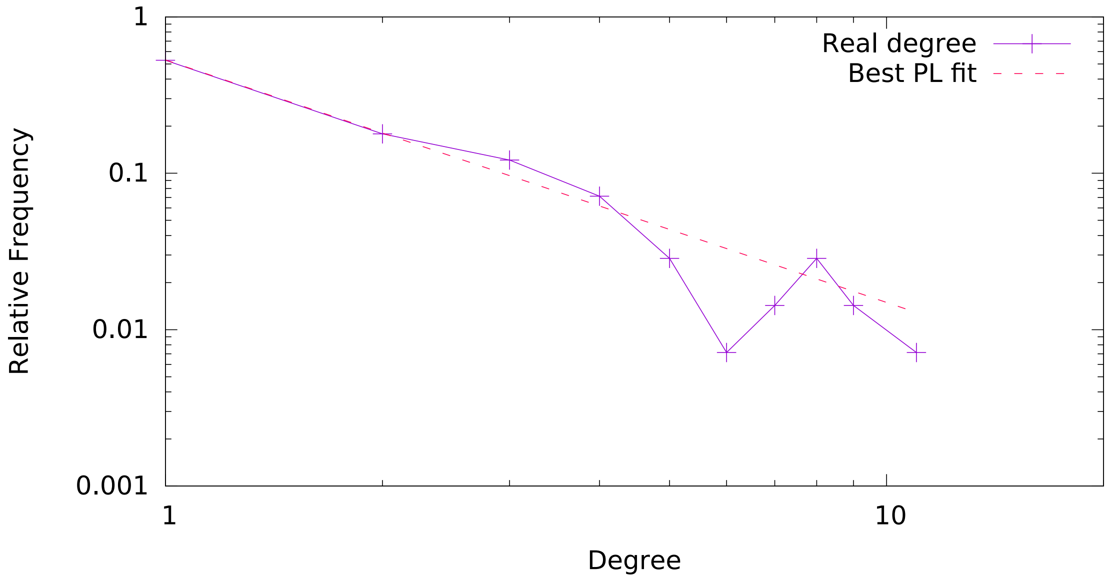


Figure 2.3: The degree distribution for ninux, and the best power-law fit $x^{-\alpha}$, $\alpha = 1.55$.

fluctuations, as in [32], the cumulative function $\mathrm{k}(k)$ is considered:

$$\mathrm{k}(k) = \int_0^k \Pi(x)dx \qquad (2.1)$$

In case $\Pi(k) \propto k$ then $\mathrm{k}(k) \propto 2$, which reported in a log-log graph should be a straight line of slope 2. Figure 2.4 report $\mathrm{k}(k)$ for various years in the two FunkFeuer network and show clearly that there is no linear trend. Indeed, fig. 2.5 confirms that for none of the years under analysis $\Pi(k)$ grows with k.

These results show that the two networks for which data is available (for the ninux network the monitoring period is too short) the growth model does not support the hypothesis of preferential attachment. As said, preferential attachment is not the only network generator that leads to scale-free networks, so next section gives also some qualitative interpretation of this result.

2.3.1. Interpretation of the results

Two features that influence the growth of a CN are: i) the limited range of wireless links ii) an upper bound on the number of incoming links.

Limited range: Wireless links are limited to a maximum length of about tens of kilometers and need to have line-of-sigh between the endpoints, while wired links do not have this limitation. Thus, a new node entering the network can not connect to any other node, and an existent node can acquire new links only from nodes placed at a distance smaller than the maximum range (which is not a fixed value and depends on a number of factors, such as the antenna type, the transmission power etc.). If the network grows in an urban area maintaining a constant density, hubs will be formed less likely than in a scale-free network.

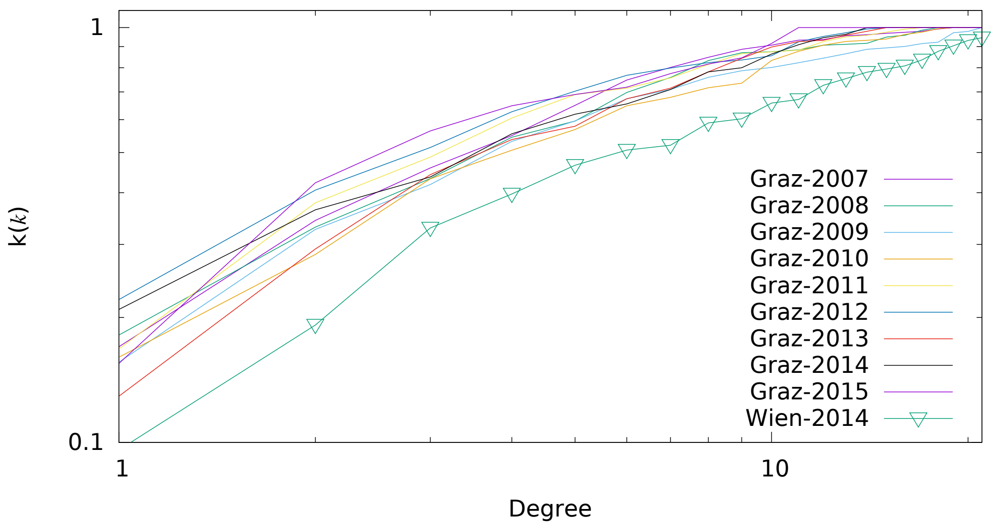


Figure 2.4: The value of k(k) in the FFGraz and FFWien networks, separately computed per each year

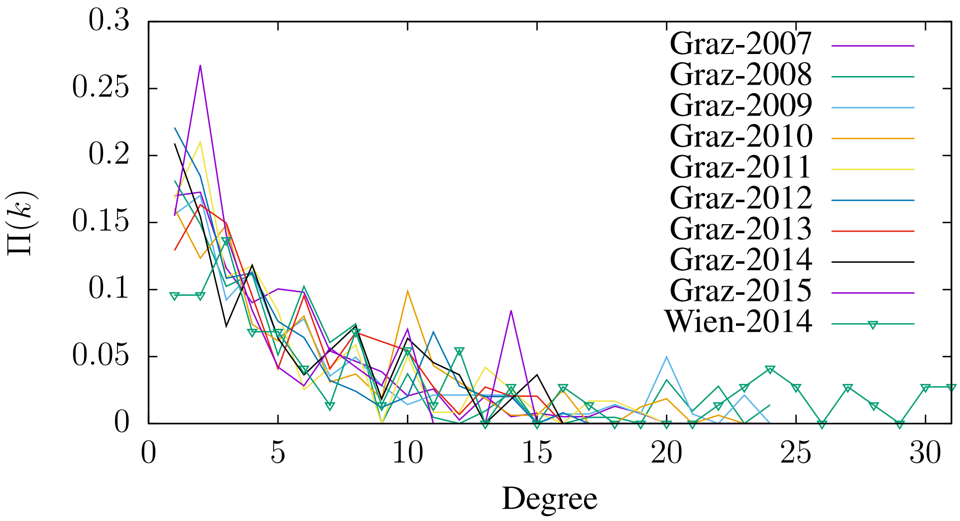


Figure 2.5: The value of $\Pi(k)$ in the FFGraz and FFWien networks, separately computed per each year

Limited maximum node degree: A wireless node can be equipped with several physical radios, but more radios require more maintenance. Mounting tens of radios, cabling them, powering them, configuring them, is costly. While wireless ISPs use trellis and pay for the maintenance, a single person typically does not have the physical space, the resources and the time to install and maintain such a complex infrastructure. Thus, node degree can not grow indefinitely.

This result confirms and extends the analysis carried on portions of the Guifi network [26] that observed that some portions of Guifi did not show a scale-free behaviour. The authors suggest that this is true for networks that cover up to a certain geographical area and it is influenced by the degree of ”planning” in the evolution of the network (planned or completely spontaneous). It must be noted that Guifi, contrarily to the networks analysed in this work is a large scale network with a mixed wired-wireless technology. This probably leads to shorten as much as possible the path to the closest gateway, and a quasi-hierarchical network design is more suitable for this task. The network thus is not anymore completely spontaneous but it develops with a mix of spontaneous growth and planning, which probably drives it towards a more hierarchical structure. This interpretation would lead to the conclusion that when the analysed networks will grow, they will also take the shape of a scale-free network but more time and research is needed to formulate a sound interpretation.

2.4. The Ownership of Ninux

Peer-to-peer organization is a key feature of ninux: since the mesh network works without introducing hierarchies and layers, the community tries to reflect this approach also in the social organization. Thus, the ninux community did not create a formal association, it does not assign formal responsibilities and does not have “roles” assigned to people. The discussions in the community are primarily carried on in the mailing lists and in weekly face-to-face meetings, and decisions are taken with a consensus-based method. Using this approach Ninux was able to build a network made of hundreds of nodes, especially concentrated in the Rome area, that serves and connects a lively community of hackers and experimenters [24]. Ninux is not the only network that uses such a distributed approach, but it is probably the one that puts the strongest emphasis on the decentralization of the network and the community.

Figure 2.6 presents the number of nodes possessed by the top-20 ninux participants, ordered by nodes owned. Over a total of 85 owners, one user possesses 17% of the nodes and the top-five people own 31% of the nodes, top-13 people own roughly 50% of the nodes, 61 people own just one node. If we exclude the first individual (that we call P_{top}), the ownership distribution is not particularly skewed, reflecting the fact that the number of owned nodes is generally limited by the number of physical locations to which the person has access (home, workplace, houses of relatives etc. . .). P_{top} owns 24 nodes and is not the owner of all the locations where the nodes are placed, he is simply a technically skilled person that very often offers his help to set up the network for newcomers. As a result, he appears to be the owner and he is the technical manager of the nodes.

Figure 2.7 shows the group betweenness centrality computed on all the nodes owned by the same person. The group betweenness centrality is the fraction of shortest paths that pass through at least one node in the group. Formally, if the network graph is a weighted graph $G(V, E)$, and $P_{i,j} = \{v_i \dots v_j\}$ is the set of nodes that constitute the shortest path from node v_i to node v_j then the group centrality of a set of nodes $S = \{v_1 \dots v_n\} \subset V$ is given by:

$$B(S) = \frac{||\{P_{i,j}\ i,j \in (1 \dots |V|)\,, i \neq j \mid S \cap P_{i,j} \neq \emptyset\}||}{||\{P_{i,j}\ i,j \in (1 \dots |V|)\,, i \neq j\ \}||} \tag{2.2}$$

where $||\cdot||$ is the size of a set. This definition is functionally equivalent to the original definition by Borgatti ad Everett [33] with two marginal differences: first, it includes also the shortest paths that use a node in S as an endpoint, second it assumes there is only one shortest path between any couple of nodes. The second assumption derives from the fact that wireless links are weighted, and thus there

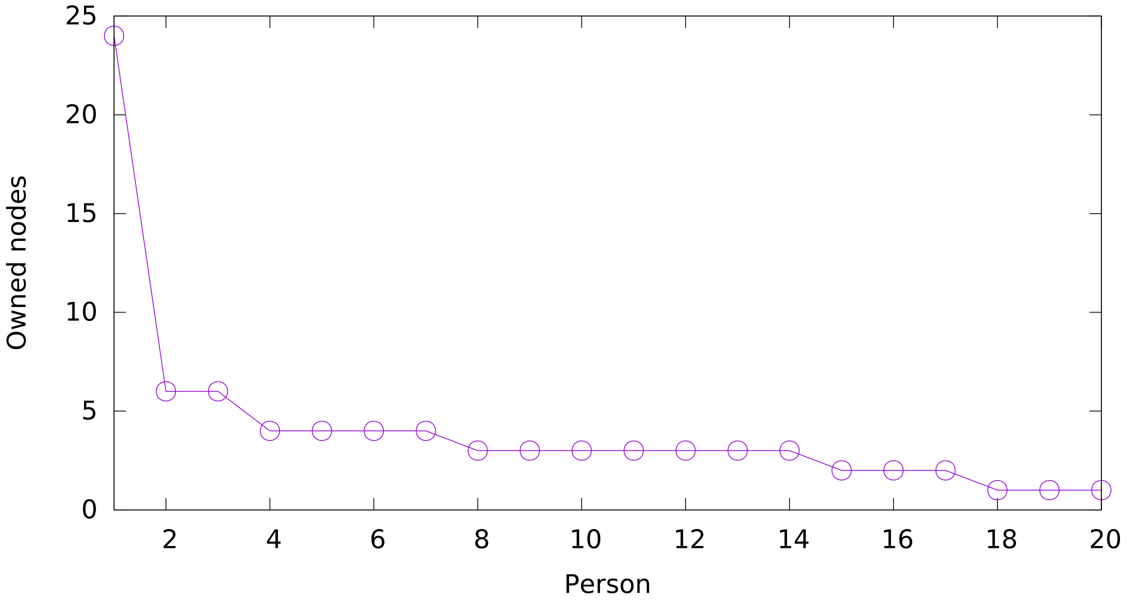


Figure 2.6: The number of nodes per user in the ninux network, top-20 users.

can hardly exist two paths with the same exact total cost. The centrality metric is computed running Djikstra's algorithm on the weighted network topology, and, without information on the traffic matrix is the best estimation of the number of traffic flows that a group of nodes can intercept. fig. 2.7 shows two metrics, the "node to node owner centrality" and the "person to person owner centrality". The first metric is exactly eq. (2.2) when S groups all the nodes of a single person. The second metric is a modified version of eq. (2.2) when $P_{i,j}$ is not computed on every couple of nodes but only on the shortest path that interconnects nodes belonging to two different people. It expresses the centrality of an owner between couples of other owners. Both metrics show that the P_{top} can potentially control between 80% and 90% of the traffic flows.

Being in between of many shortest paths gives to P_{top} an advantage position to control the network. He would be able to spy on a large quantity of the traffic and to filter it. While there is no reason to believe the person was actually enforcing those behaviours, the important observation here is that such a large predominance in the network topology gives to one single person a strong influence and a high decision power.

2.4.1. Owner Robustness

One classical way to inspect the robustness of network graphs is to remove some nodes in the topology and check the connectedness of the remaining network [34]. A network is fragile if removing a small amount of nodes it is split in many small separated networks. Let S_i be the set of nodes owned by owner i, and let $R_G(S_i)$ be the size of the largest connected component when S_i is removed from the network. $R_G(S_i)$ is a robustness metric, the closer to $||V||$, the better. With a little abuse of notation we call $R_G(S_i)$ the robustness of owner i (instead of calling it "the robustness of the network to the failure of all the nodes in S_i"). A related metric is the number of disconnected components left in the network when S_i is removed, which we call the fragility of i: $F_G(S_i)$.

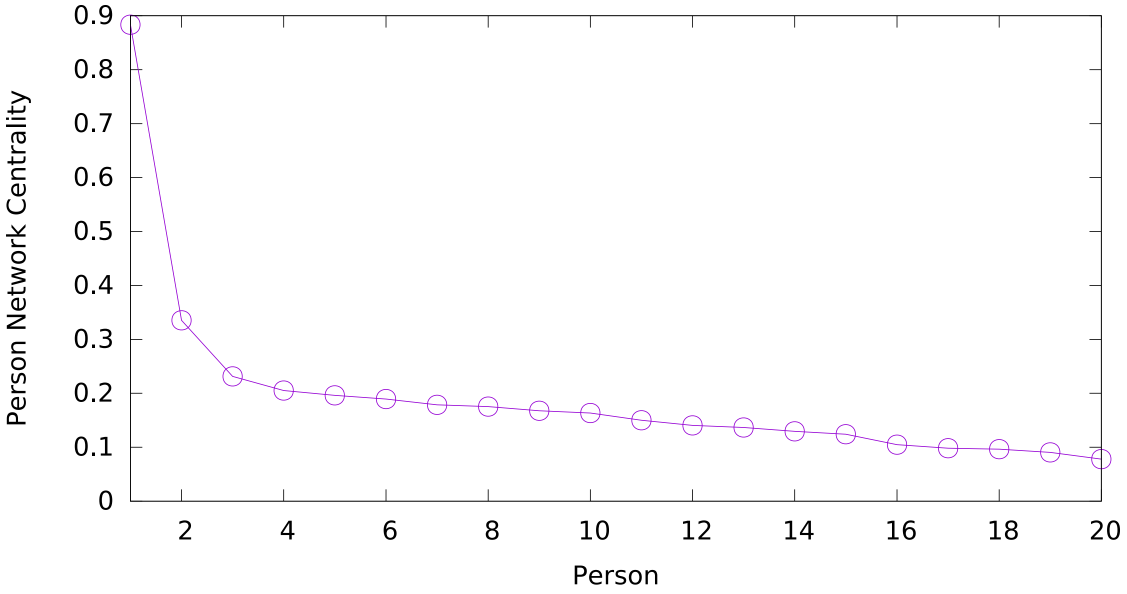


Figure 2.7: The owner centrality for the participants to the ninux network, top 20 users.

We have already shown that community networks are not in general robust to targeted attacks [13], it interesting to observe what happens if one person leaves the network or turns off all its nodes. figs. 2.8 and 2.9 show both robustness and fragility of the owners ordered by the number of nodes owned (as in fig. 2.6).

Again it is clear that there is one person that represents a single point of failure of the network. When the nodes belonging to P_{top} are removed the main connected component is reduced to less than half the size of the original network and the remaining nodes are distributed in more than 30 isolated components.

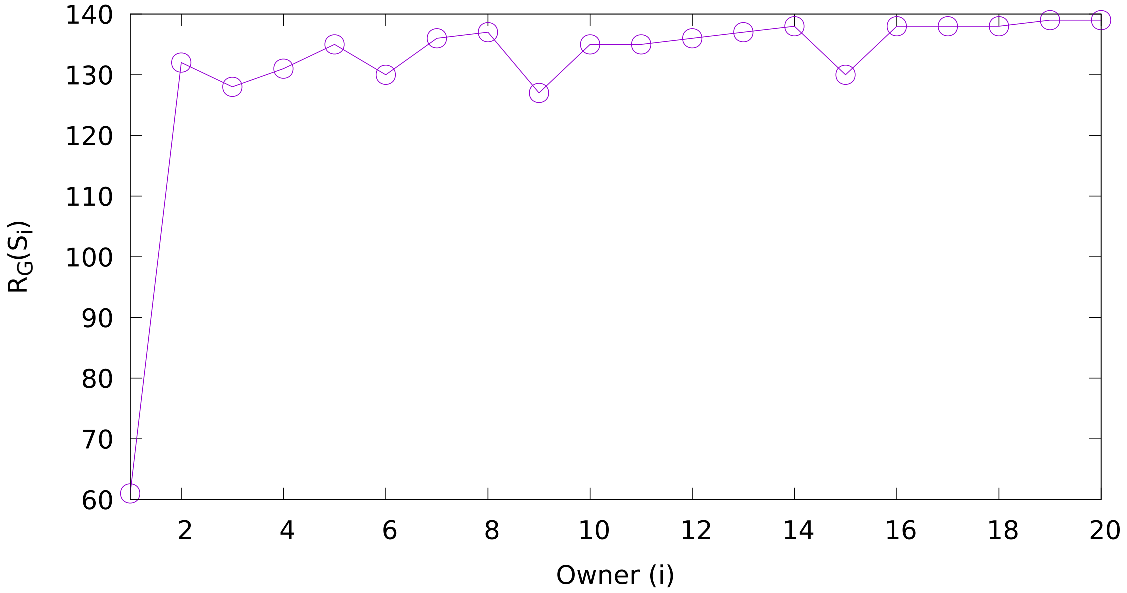


Figure 2.8: The size of the largest connected component remaining after the removal the set S_i of the nodes belonging to owner i.

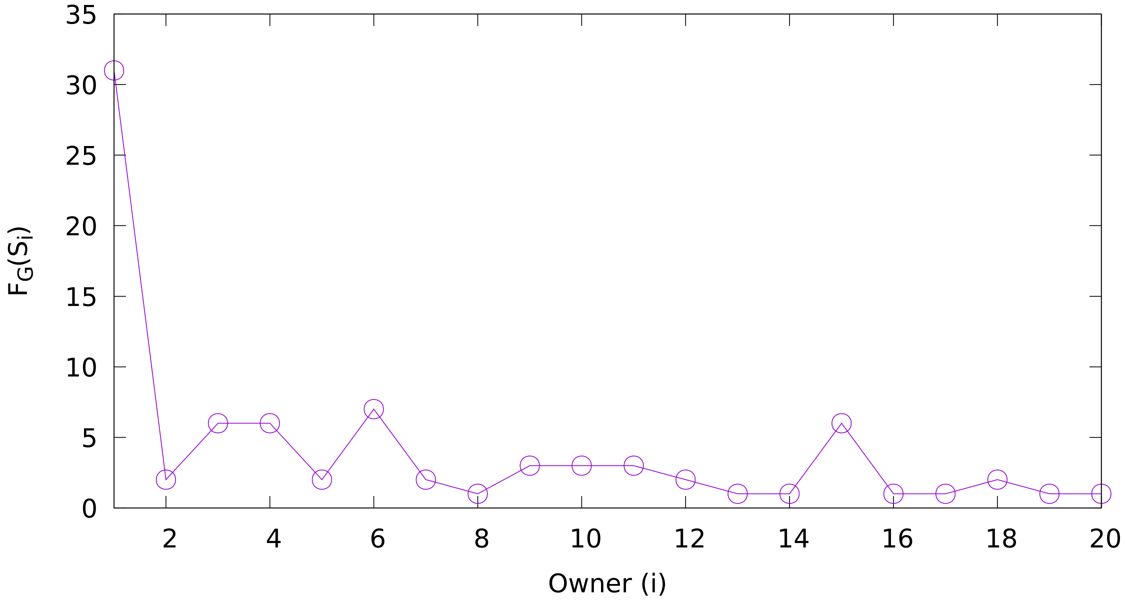


Figure 2.9: The number of disconnected components after the removal of the set S_i of the nodes belonging to owner i.

2.5. Analysis of Ninux's Social Network

The analysis of the mailing list messages helps understanding who are the individuals that lead the discussion inside the community. Two metrics defined in the literature have been chosen for this task [35]. The first is the normalized number of answered email per user: given a number X of total messages that reply to some other message, and being x_i the number of replies to a message sent by the ith person, $R(i) = \frac{x_i}{X}$ is the relevance metric shown in fig. 2.10. This is a basic metric that assumes that people that receive a high number of replies are able to generate interesting discussion topics, thus are considered important in the community.

Figure 2.10 shows that the relevance to the mailing list is not equally distributed among the participants, a very small number of people lead the discussion. The cumulative distribution in fig. 2.11 shows that as less as 6 people receive 50% of the overall answers. This is not uncommon in many mailing lists, for instance, in open source projects the distribution of the participation to mailing lists is generally very skewed with a minority of people leading the discussion [36]. Unfortunately ninux does not represent an exception.

The second metric is the centrality of a person in the mailing list social graph. The social graph is an undirected graph $G(V, E)$ in which every node v_i is a person in the mailing list and there is an unweighted edge between two nodes v_i, v_j if person v_j ever answered to person v_i (or vice-versa). Mailing list centrality is computed on the social graph for v_i as in eq. (2.2) when $S = \{v_i\}$. Betweenness centrality on mailing lists is used to understand who is able to make other people join the same discussion, so that he/she can facilitate the flow of information in the community. Again, fig. 2.12 shows that there is a small number of people connecting all the other participants, and one in particular whose centrality is at least the double of the others.

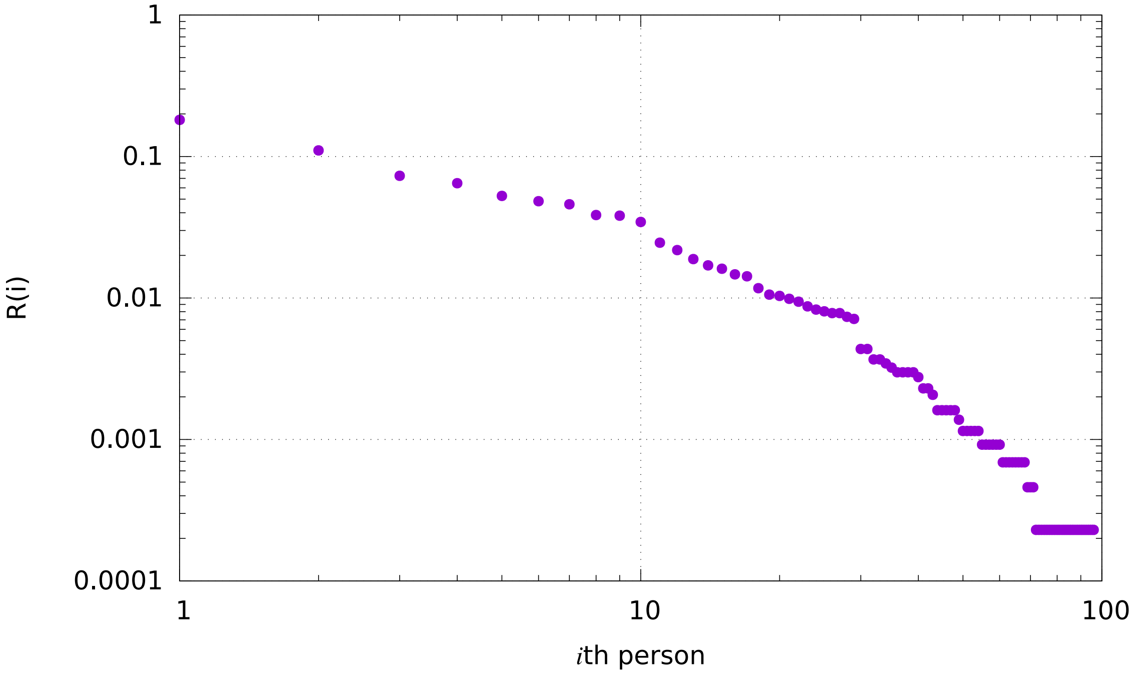


Figure 2.10: The fraction of answered emails on the total in the mailing list.

Another layer of analysis is given by the identification of communities in the mailing list graph. Among the several available community identification algorithms the Louvain method [37] was cho-

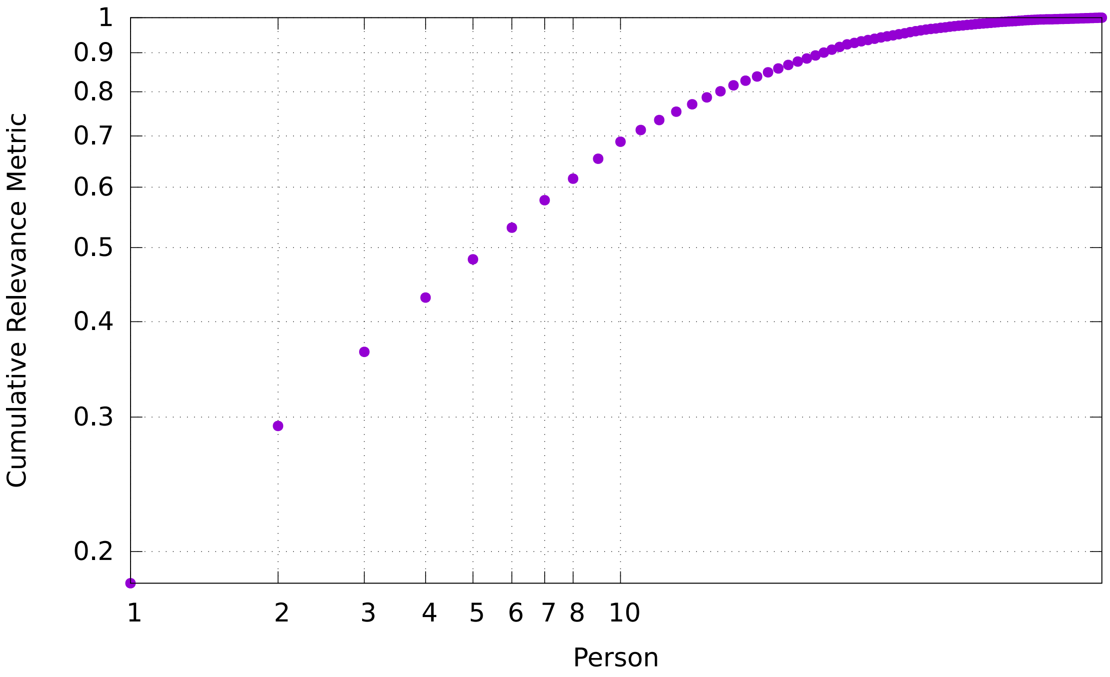


Figure 2.11: The cumulative distribution of answered emails on the total in the mailing list.

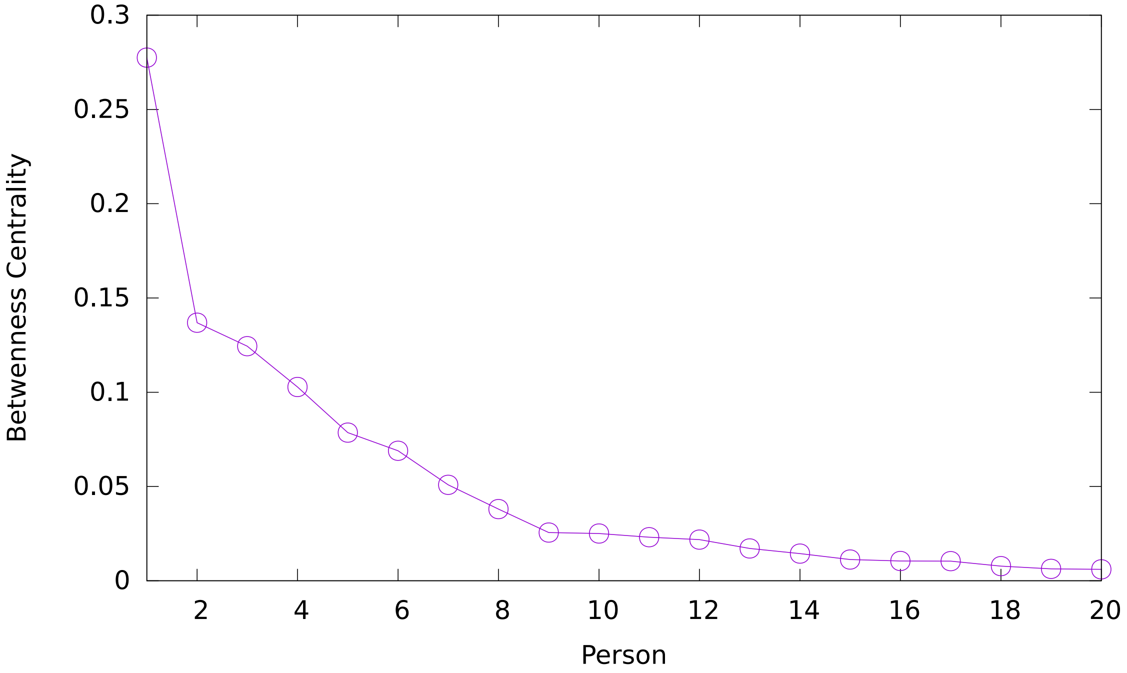
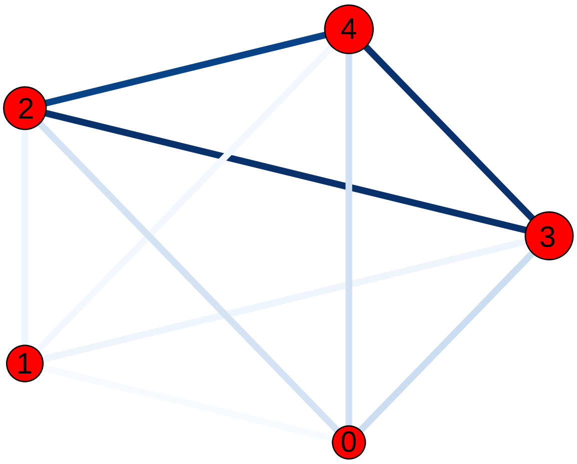


Figure 2.12: The ranked centrality of the top 20 participants in the ninux mailing list.

sen and applied to the ninux mailing list. The algorithm identified 9 communities, among which 4 made of a single user, the partition modularity is 0.156. fig. 2.13a shows the interaction graph between the communities of more than one person. The size of each community and the strength of each link is reported in figs. 2.13b and 2.13c. The graph shows that the ninux mailing list is quite "compact", meaning that there are only five communities, three of which include 80 users and are very well connected with each other. The modularity is not very high. Apart from a small set of 4 source email addresses, everybody belongs to some community.



(a)

Com.	Size
0	9
1	13
2	21
3	29
4	30

(b)

	0	1	2	3	4
0	47	4	129	155	143
1	4	51	30	38	22
2	129	30	541	682	634
3	155	38	682	627	679
4	143	22	634	679	569

(c)

Figure 2.13: (a) The communities identified in the ninux mailing list. Circle size reflects community size, edges gradient represent the relative strength (number of exchanged emails) of the connection: deep-blue strong connection, light blue weak connection. (b) The size of each community. (c) Number of emails exchanged between communities.

2.6. Matching the Communications and the Social Network

Figure 2.14 reports the percentage overlap on the two betweenness rankings from fig. 2.7 and fig. 2.12. The percentage overlap gives a measure of the correlation between the two rankings. Given a family of sets B_i and the respective ordering functions $o_i(v)$ on their elements, we call B_i^k the first k element of

B_i ordered by $o_i(v)$: $B_i^k = \{v | v \in B_i, o_i(v) \leq k\}$. Given two sets B_1 and B_2 the percentage overlap $p(k)$ is a function of k that shows the percentage of elements present in both sets when considering only the first k elements:

$$p(k) = \frac{100}{k} \times ||B_1^k \cap B_2^k|| \tag{2.3}$$

Figure 2.14 shows two fundamental points: the first is that P_{top}, the person that owns more nodes and has the highest person network centrality is the same one that has the highest centrality in the social graph. This confirms that the distributed community management of ninux does not itself avoid the presence of single points of failure in the community network. There is one person that contributes to the growth of the network, and to the mailing list discussion in a way that gives him a tremendous power to steer the direction of the community. The second point evidences a different, and more encouraging trend. If we exclude P_{top} the correlation between the communication and the social network centrality is not extremely evident, $p(10) = 20\%$ and $p(20) = 35\%$. Therefore it seems that there is diversity between the owner of the most critical nodes and the leaders of the discussion in the mailing list. Thus, the general idea that the use of free to access communication and discussion tools guarantees plurality and participation is only partly matched by reality.

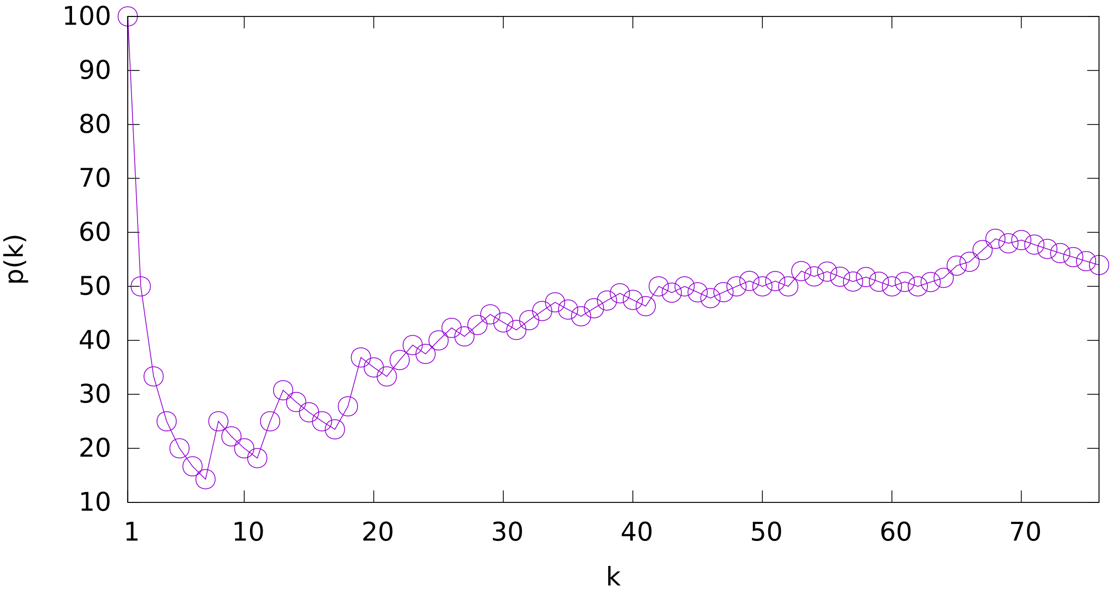


Figure 2.14: The percentage overlap metric computed on the ranked mailing-list and group node centrality.

Another encouraging element raises from fig. 2.15 that reports the number of owners grouped for the community they belong to among the top 5, 10, 15, and 20 node owners. Communities are ordered for their size from the largest to the smallest. The figure shows that the distribution of the top owners per community is not particularly skewed towards one community. At least three communities are present in all the bars so there is not a single clique of users that dominates the communication network and the mailing list discussion. Another element is that the owners with more nodes actually participate to the mailing list. Even if only 44 owners over 85 are present in the mailing list, only 20% of the top 10, 15, and 20 owners do not participate to the mailing list.

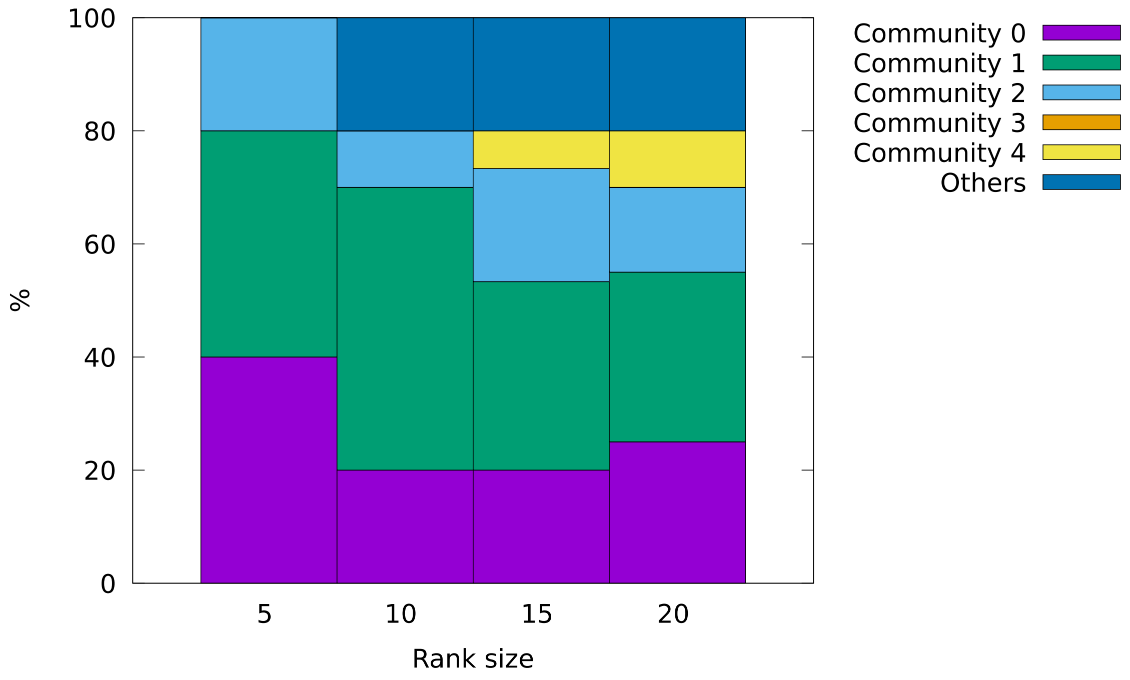


Figure 2.15: For the top 5, 10, 15, and 20 owners of network nodes, the percentage of belonging to some community

2.7. Interpretation of the results

The distribution of the ownership, and thus the person centrality shows that, albeit the goal of the ninux community is to build a technically and socially decentralized network, the results diverge from the goal. In 2014 one person in ninux managed to control a sufficient number of nodes so as to be able to control the whole network, and to represent a single point of failure. The same person, given his technical skills was a central person in the social network of the community, so he had an influential voice in the discussions. P_{top}, without being one of the founders of the community, become quickly a reference point for the community. He strongly influenced the technical choices and become a hidden leader of a group of people. Direct interactions with the community revealed that the centrality of this person in the network and in the discussion (both in the mailing list and in the live meetings) led to conflicts and eventually to the fracture of the community in separate factions. Several people felt that the network was evolving in a way that was too fast and not enough participated, and felt excluded by the decision process.

Indeed, direct discussion engaged with people in the community revealed that this person left the community in 2015 and the nodes he managed started to fail and disconnect entire areas. At the time of writing the main component of the network is made of 87 nodes, and the other nodes were disconnected from the main component. At the same time, the quantity of email exchanged in the mailing list dropped down in the last two years, In conclusion, the approach of the ninux community for the decentralization of the technical and social network was not successful since the network had a single point of failure represented by P_{top}.

However the situation changes, when excluding P_{top} from the analysis. Figure 2.6 shows that the maximum number of owned nodes is generally capped by the amount of physical locations that the users have access to, which intrinsically limits the chances of some individuals to take over the network.

Also, even if the social network metrics show that the relevance of the participants to the mailing list is not evenly distributed (this is indeed pretty common in many mailing list [36]) the correlation between the most relevant node owners and the most relevant members of the mailing list is low. This means that people participate to the community in diverse ways, with the construction of new nodes or through rising discussion topics.

A simple solution to this problem would be to prevent people to manage nodes in physical location they do not own. This way, Wi-Fi range limitation would not allow a single person to be too central, and thus too critical for the network economy. This would change the nature of a CN which instead, to be able to grow, must be participated not only by individuals but also by associations and small businesses that can be physically located in several places. A better solution is to reassign the ownership of nodes or to share the management credentials of nodes among several people as next section proposes.

2.8. Node re-assignment procedure

Once the community accepts that a person can administer one node without owning the corresponding physical location, we can leverage on this to redistribute the responsibility of some nodes from the original owner to somebody else, in order to reduce the importance of each single person. In this section an algorithm is introduced whose goal is to raise the minimum $R_G(S_i)$ beyond a certain threshold.

Re-assigning the control of nodes can be done with any node-labelling algorithm that will maximise the fair distribution of nodes among the people in the community. Of course, that algorithm would not consider practical constraints, such as the chances that the old owner would not trust the new owner, or that for the new owner the physical access to the node could not be easy. In practice, some real-word constraints must be introduced to make the resulting re-assignment practically useful. To do that, let's first introduce some symbols. Let $C(i)$ be the community of node i in the social graph, and S_i the nodes owned by owner i. Let S be the set of all the nodes in the communication network and C be the set of all the node owners. When re-assigning a node v from owner i to node j the proposed assignment scheme is based on the following constraints:

- $j \in C(i)$: If this is not possible, then j will be chosen iteratively from the next community that has the strongest link with $C(i)$.
- $S_j \neq \varnothing$: In order for j to have enough technical skills he must be the owner of at least one node before re-assignment.
- The probability of owner $j \in C(i)$ of being assigned the management of node $v \in C(i)$ increases linearly with the minimum distance in number of hops from v to any node in S_j. More formally, if $d_{ij} = min\{||P_{ij}|| \, \forall \, i, j \, | \, v_i \in S_i \wedge v_j \in S_j\}$ then for the probability $P(i)$ and $P(k)$ of owner j and k to be assigned v it is true that $P(j) = P(k)\frac{d_{ik}}{d_{ij}}$. In other words, there is a bias in reassigning a node to owner j if owner j owns a node that is physically close to v. The rationale behind this choice is that if physical maintenance is needed, it is more likely that j will be close to v. We approximate physical distance with the number of hops, but this strategy can be improved if information on the geographic coordinates of nodes is available.

fig. 2.16 describes an heuristic algorithm that takes the least robust owner and reassigns his/her nodes up to when his/her robustness is higher than a threshold T. The procedure must be repeated up to when $R_G(S_i) > T \, \forall \, i$.

The first challenge is to define T, which can not be arbitrarily high, but depends on the network structure. To achieve this, the algorithm computes the robustness $R_G(\{v\})$ for every node v in the network and finds the node v_l with the lowest $R_G(\{v_l\})$. It is intuitive that for any person i so that $v_l \in S_i$, $R_G(S_i) \leq R_G(\{v_l\})$, so initially we set $T = R_G(\{v_l\}) - 1$. If at the end of the execution a solution can not be found, T is decremented and the algorithm is run again. Another important observation is that when computing $R_G(S_i)$ the contribution of leaf nodes must be omitted. Let L_i be the number of leaf nodes owned by node i, with $L_i \subset S_i$. There are pathological cases in which the lowest value of $R_G(S_i) < T$ corresponds to an owner i for which $S_i = L_i$. In this case the reassignment of nodes will start reassigning all the nodes of i one by one up to when $R_G(S_i) > T$. It makes no sense to redistribute the ownership of a leaf node since its failure only affects the owner of the node. So, when comparing $R_G(S_i)$ with T, $R_G(S_i)$ is increased of the size of $||L_i||$.

Given these premises the algorithm in fig. 2.16 does the following:

Lines 1-4: identify the owner i and the node n_l with the lowest robustness.

Lines 5-7: define S_i and L_i.

Line 9: start the re-assignment of non-leaf nodes

Line 15: pick a random person in $C(i)$, with a bias on close-by people.

Line 18-19: if no person can be chosen, break (jumps to line 38)

Line 20-24: test if re-assignment is feasible. If the new owner after re-assignment has robustness below threshold, blacklist him.

Line 26-34: checks if after the re-assignment of v, $R_G(S_i)$ is still below the threshold. If not, exit from main loop.

Line 38-42: if the re-assigning process in unsuccessful, decrement the threshold, undo changes and loop again.

The main loop starting at line 9 is executed once every time T is decremented, so at most $T < ||V||$ times. The loop starting at line 11 runs at most $||S_i|| < ||S||$ times, while the inner loop starting at line 14 runs at most $||C(i)|| < ||C||$ times. The most complex operation in the loop is computing $R_G(S_i)$ that requires the computation of all Dijkstra's trees which is an operation with polynomial complexity on $||S||$. The algorithm complexity thus remains polynomial on $||C||$ and $||S||$, and it instantly finds a solution on normal hardware for the ninux network. Further optimizations can be done, but are out of the scope of this work, whose goal is to showcase the feasibility of such an approach.

For ninux, the algorithm produced the reassignment of 6 nodes from P_{top} to 6 different people, fig. 2.17 reports the corresponding graph of $R_G(S_i)$ for the top-20 owners after re-assignment and shows that the minimum robustness is strongly increased.

```
R_o = sort_owners_robustness() # returns a sorted list of R(S)
R_n = sort_nodes_robustness() # returns a sorted list of (node, R(node))
T = R_n[0] # set T to the lowest node robustness
least_robust_owner = R_o[0][0]
sorted_owned_nodes = get_sorted_nodes_by_owner(least_robust_owner)
# returns a list of nodes for an owner, sorted by their robustness
leaf_nodes = get_leaf_nodes(sorted_owned_nodes)
exit_loop = False
while not exit_loop: # main loop
  reassigned_nodes = []
  for node in sorted_owned_nodes.remove(leaf_nodes):
    # loop on non-leaf nodes
    black_list = []
    while True: # inner while loop
      new_friend = get_random_friend(node, black_list)
      # return a random person in the communitiy of the owner
      # of node, excluding the black_list
      if not new_friend:
        break  # no one can receive this node. break inner while loop
      if not test_reassign(node, new_friend, T):
        # temporarily reassing the node to new_friend, recompute
        # its robustness, return False if new_friend
        # is himself breaking the T, keep looping
        black_list.append(new_friend)
      else:
        reassign_node(node, new_friend)
        # reassign the ownership to new_friend
        reassigned_nodes.append(node)
        # keep track of reassigned nodes
        new_owned_nodes = sorted_owned_nodes.remove(node)
        new_robustness = compute_robustness(new_owned_nodes)
        # recompute the robustness
        if new_robustness + len(leaf_nodes) > T:
          exit_loop = True # will exit main loop
        break  # exit inner while loop, keep looping in the for loop
    if exit_loop:
        break  # for loop
  # Failed to reassign nodes: must decrement T,
  # reset all done and try again
  T.decrement(1)
  for node in reassigned_nodes:
    reassign_node(node, least_robust_owner)
return reassigned_nodes
```

Figure 2.16: Re-Assigning heuristic.

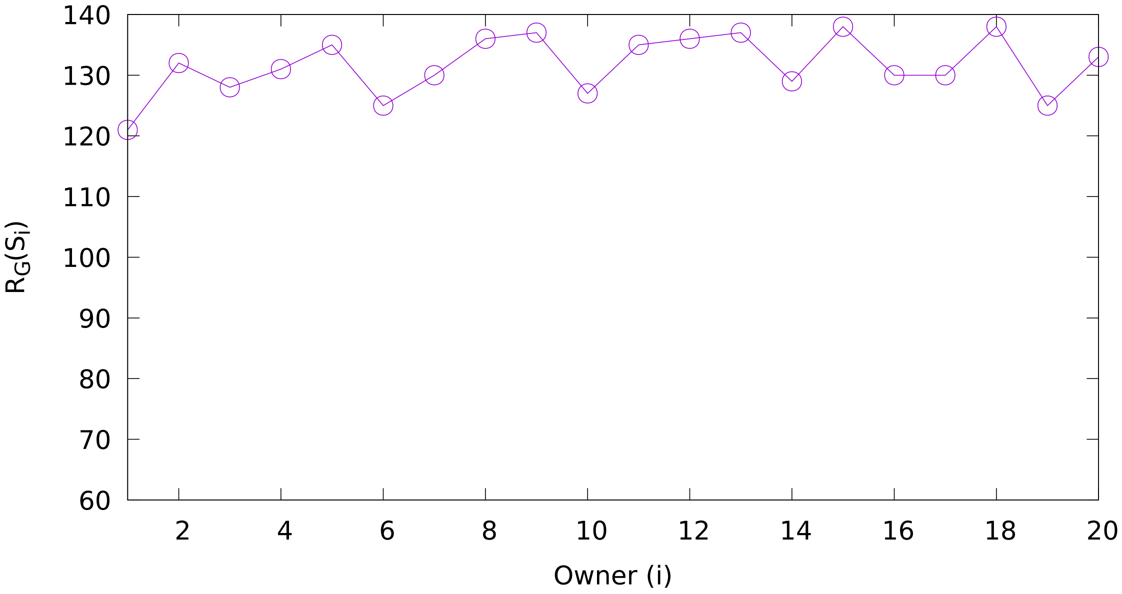


Figure 2.17: The size of the largest connected component remaining after the removal of all the nodes belonging to a person, after the node re-assignment process.

3. Conclusions and the Way Forward

Some CNs have a very clear social vision and propose a networking model that is distinctly different from the one offered by commercial ISPs. In the light of the growing debate on network neutrality and network access, CNs represent a promising alternative and/or complementary model. The novelty of such model lies both in its technical organization as a mesh network, and in the governance of the network, that tends to be horizontal and participatory. Nevertheless, a community organized around a distributed network and open social networking instruments may anyway develop in unbalanced ways and give rise to single points of failure. The work in netCommons T2.4 is aimed at understanding to what extent these threats are realized in real community networks, and this deliverable reports on the related activities carried out in three mid-size networks daily used by hundreds of people.

The main observation is that even if, theoretically, the distributed nature of a CN favors a more balanced distribution of importance across users, in practice, and due to the spontaneous evolution of the network, neither the network topology nor the social network that emerges out of the CN users' interactions on the mailing list exhibit flat balanced structure.

The implication is that it may be worth monitoring this spontaneous evolution of the network and complementing it with a set of instruments that let the community understand the direction it is taking and act accordingly. We propose an approach to monitoring the communication and the social network, which identifies single points of failure with the help of graph-theoretic concepts. The proposed metrics and the overall methodology can be used by any network to detect the emergence of problematic situations before they become critical for the network sustainability. Furthermore, and as a countermeasure against such situations, we devise a heuristic algorithm for re-assigning the ownership of selected nodes so that the robustness of the communication network is increased.

Ideally, one would also like to re-assign broker positions in the social network graph. This is obviously non-trivial since the position in the social graph depends on human behaviour. Nevertheless, the community could encourage the participants to take the lead in some actions or the responsibility for specific themes in order to promote their role in the discussions that take place in the mailing list. This would artificially change the shape of the social network graph and induce higher balance in the social network structure, in the same way that the proposed node ownership re-assignment heuristic did for the network topology.

3.1. The Next Steps

On the 26th and 27th of November 2016, the work carried out in T2.4 was presented to the ninux community, while they held their national meeting in Florence. Leonardo Maccari from UniTn contributed to the organization of the meeting and participated in it[1]. It was a precious opportunity to serve two significant purposes. The first one was to share with the community the data and the general conclusions that were drawn in this analysis. The community confirmed that the analysis reflects the state of things and appreciated that such analysis could be carried to understand the way that the

[1] See netcommons website for a brief report of the meeting http://netcommons.eu/?q=content/netcommons-ninuxday-meeting-ninux-community-network

community itself evolves. The second purpose was a proposal to integrate the developed methodologies into the Ninux monitoring infrastructure. Ninux uses a mapserver, a web application that displays the network map and other performance details. The current version of the mapserver will be soon replaced with a new version[2]. The software developed in T2.4 will be integrated in the new mapserver release and will provide the community with available data to understand the evolution of their network.

As a first step of the integration process, we will embed network robustness functionality into the mapserver tool and give information about the fragility of the network graph. We will also extend the node fragility concept to the owner fragility counterpart so that the community can be aware of the creation of single points of failure. An appropriate visualization strategy will be proposed for the mapserver in order to immediately understand what are the critical parts of the network infrastructure that deserve attention.

The integration of information from the mailing lists will be exhaustively discussed with the community. This is a complex task having strict privacy requirements since the mailing lists are not generally open to non-subscribers. Moreover, while modifications of the network graph can be planned and realized, modifications of the interaction graph of the mailing list can not be done mechanically. The community needs to find a way to integrate the newcomers, enlarge the discussion topics in order to be more inclusive and at the same time remain cohesive. This process needs guidance to understand what are the best instruments that can help maintain a participatory and horizontal governance, which is the goal of Work Package 1 (WP1). A further suggestion raised by the community is to understand the feasibility of integrating the proposed metrics in available open-source community monitoring software such as the Grimoirelab[3]. Grimoirelab is a platform to monitor the "health" of an on-line free software development community; hence, it is tailored to software production and not to community networks. In the second year of T2.4 we will assess whether this platform can be extended with metrics that are useful for CNs and consider ways to contribute to its development.

Another line of work will be the characterization of other networks, both topologically and socially. One of the largest sources of information that is today unused, is the information published by the Freifunk CN. Freifunk is made of literally hundreds of networks, whose size ranges from few nodes to few thousands. Structured information about the network is collected via the so-called Freifunk API, developed by the community. Extending the work carried out on the FunkFeuer and Ninux topologies to the Freifunk network will corroborate the current findings and give more material for further research.

Finally, on a more theoretical note, we will seek to improve upon the heuristic algorithm proposed and implemented for improving the robustness properties of the CN topology. More specifically, we will try to formulate the problem as an online optimization problem, instantiated each time a new node emerges that wants to connect to the network. The aim will be to derive optimal recommendation policies that will encourage the user to connect with specific nodes each time two or more nodes are within communication range.

[2] The current mapserver can be found at http://map.ninux.org and the new one, currently under development, is currently hosted at http://ninux.nodeshot.org

[3] See http://grimoirelab.github.io/

Bibliography

[1] L. Maccari, "On the Technical and Social Structure of Community Networks," in *The First IFIP Internet of People Workshop,IoP*, 2016.

[2] P. Baran, "On distributed communications networks," *IEEE transactions on Communications Systems*, vol. 12, no. 1, pp. 1–9, 1964. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1088883

[3] W. Waites, J. Sweet, R. Baig, P. Buneman, M. Fayed, G. Hughes, M. Fourman, and R. Simmons, "RemIX: A Distributed Internet Exchange for Remote and Rural Networks," *arXiv preprint arXiv:1603.08978*, 2016. [Online]. Available: http://arxiv.org/abs/1603.08978

[4] S. Bafna, A. Pandey, and K. Verma, "Anatomy of the Internet Peering Disputes," *arXiv:1409.6526 [cs]*, Sep. 2014, arXiv: 1409.6526. [Online]. Available: http://arxiv.org/abs/1409.6526

[5] H. G. Schulzrinne, "Key parameters for universal internet access: Availability, affordability and relevance," in *Dagstuhl Reports*, vol. 4, no. 11. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

[6] R. D. Doverspike, K. K. Ramakrishnan, and C. Chase, "Structural overview of ISP networks," in *Guide to Reliable Internet Services and Applications*. Springer, 2010, pp. 19–93. [Online]. Available: http://link.springer.com/10.1007/978-1-84882-828-5_2

[7] N. Hurtig, J. Brown, and T. Johansson, *Creating an Internet Service Provider: Design and Implementation of a small scale ISP*. Saarbrücken: LAP LAMBERT Academic Publishing, Nov. 2010.

[8] P. Chanclou, Z. Belfqih, B. Charbonnier, T. Duong, F. Frank, N. Genay, M. Huchard, P. Guignard, L. Guillo, B. Landousies *et al.*, "Access network evolution: optical fibre to the subscribers and impact on the metropolitan and home networks," *Comptes Rendus Physique*, vol. 9, no. 9, pp. 935–946, 2008.

[9] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi, "Detecting bittorrent blocking," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008, pp. 3–8. [Online]. Available: http://doi.acm.org/10.1145/1452520.1452523

[10] D. Miorandi, I. Carreras, E. Gregori, I. Graham, and J. Stewart, "Measuring net neutrality in mobile internet: Towards a crowdsensing-based citizen observatory," in *2013 IEEE International Conference on Communications Workshops (ICC)*, June 2013, pp. 199–203.

[11] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu, "Glasnost: Enabling end users to detect traffic differentiation," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 27–27. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855711.1855738

[12] B. Schneier, *Data and Goliath: The hidden battles to collect your data and control your world*. W. W. Norton & Company, 2015.

[13] L. Maccari and R. L. Cigno, "A week in the life of three large wireless community networks," *Ad Hoc Networks*, vol. 24, Part B, no. 0, pp. 175 – 190, 2015.

[14] S. Crabu, F. Giovanella, L. Maccari, and P. Magaudda, "Hacktivism, infrastructures and legal frameworks in community networks: The italian case of ninux.org," *Journal of Peer Production*, to appear in 2016.

[15] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, 1999.

[16] ——, "Network science," 2016.

[17] R. Albert, H. Jeong, and A.-L. Barabasi, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, Jul. 2000.

[18] A. Guadamuz, *Networks, complexity and internet regulation scale-free law*. Edward Elgar Publishing, 2011.

[19] W. Willinger, D. Alderson, and J. C. Doyle, "Mathematics and the internet: A source of enormous confusion and great potential," *Notices of the AMS*, vol. 56, no. 5, 2009.

[20] R. Michels, *Political parties: A sociological study of the oligarchical tendencies of modern democracy*. Hearst's International Library Company, 1915 (English Translation).

[21] P. S. Tolbert and S. R. Hiatt, "On organizations and oligarchies: Michels in the twenty-first century," 2009.

[22] A. Shaw and B. Mako Hill, "Laboratories of oligarchy? how the iron law extends to peer production," *Journal of Communication*, vol. 64, no. 2, pp. 215–238, 2014.

[23] Y. Benkler and H. Nissenbaum, "Commons-based peer production and virtue*," *Journal of Political Philosophy*, vol. 14, no. 4, pp. 394–419, 2006. [Online]. Available: http://dx.doi.org/10.1111/j.1467-9760.2006.00235.x

[24] L. Maccari, "An analysis of the Ninux wireless community network," in *The Second International Workshop on Community Networks and Bottom-up-Broadband (CNBuB)*, 2013.

[25] L. Cerda-Alabern, "On the topology characterization of Guifi.net," in *IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2012.

[26] D. Vega, R. Baig, L. Cerdà-Alabern, E. Medina, R. Meseguer, and L. Navarro, "A technological overview of the guifi.net community network," *Computer Networks*, vol. 93, pp. 260–278, 2015.

[27] C. Barz, C. Fuchs, J. Kirchhoff, J. Niewiejska, and H. Rogge, "OLSRv2 for Community Networks: Using Directional Airtime Metric with external radios," *Computer Networks*, vol. 93, Part 2, pp. 324–341, Dec. 2015.

[28] L. Cerda-Alabern, A. Neumann, and L. Maccari, "Experimental Evaluation of BMX6 Routing Metrics in a 802.11an Wireless-Community Mesh Network," in *3rd International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug. 2015.

[29] R. Baig, R. Roca, L. Navarro, and F. Freitag, "Guifi.Net: A Network Infrastructure Commons," in *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development ACMDev*. ACM, 2015.

[30] J. Kos, M. Milutinovic, and L. Cehovin, "nodewatcher: A substrate for growing your own community network," *Computer Networks*, vol. 93, pp. 279–296, 2015.

[31] D. Vega, R. Meseguer, and F. Freitag, "Analysis of the Social Effort in Multiplex Participa-

tory Networks," in *Economics of Grids, Clouds, Systems, and Services*, ser. Lecture Notes in Computer Science. Springer International Publishing, Sep. 2014, no. 8914.

[32] H. Jeong, Z. Néda, and A.-L. Barabási, "Measuring preferential attachment in evolving networks," *EPL (Europhysics Letters)*, vol. 61, no. 4, p. 567, 2003.

[33] M. G. Everett and S. P. Borgatti, "The centrality of groups and classes," *The Journal of mathematical sociology*, vol. 23, no. 3, 1999.

[34] A.-L. Barabási, *Network science*. Cambridge University Press, 2016.

[35] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan, "Mining Email Social Networks," in *Proceedings of the 2006 International Workshop on Mining Software Repositories*. ACM, 2006.

[36] S. L. Toral, M. R. Martínez-Torres, and F. Barrero, "Analysis of virtual communities supporting OSS projects using social network analysis," *Information and Software Technology*, vol. 52, no. 3, pp. 296–303, Mar. 2010.

[37] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10.

A. Appendix

A.1. Open Source Code

The source code needed to perform the network analysis is available in the official netCommons github repository[1], and it is, as any Open Source projects an ongoing work. The code which is associated as integral part to this deliverable is tagged D2.5 and is included in this deliverable in the file `community-networks-monitoring-tools.tar.gz`. This file contains not only the developed code, but also submodules that are not normally downloaded from github and make the installation of the code practically stand-alone.

The code is realized in python and split in the three folders. The first module contains code necessary to perform the "From:" field aggregation, and in general, to analyse emails. The second module performs centrality analysis on the mailing lists, and the third module contains code that is necessary to compute centrality and robustness analysis.

A.2. Open Data

The graphs used in the network analysis are available in the `testdata/` folder of the git repository. The original data-set for the mailing list analysis can not be published, being it a repository of a private mailing list. The anonymized social network graph and the anonymized ninux topology annotated with node owners is also present in the repository. As the project progresses we will define a suitable policy (license and repository) to publish open data and collect the data used in T2.4 in the official repository.

[1] See https://github.com/netCommonsEU/community-networks-monitoring-tools/

netCommons

The netCommons project

February 24, 2017

netCommons-D2.5-1.0

Horizon 2020