European Research
7th Framework Programme

CONNECTED COMMUNITIES
CONFINE

Project title: Community Networks Testbed for the Future Internet.

Experimental research on testbeds for community networks (year 2)

Deliverable number: D.4.2

Project Acronym: CONFINE
Project Full Title: Community Networks Testbed for the Future Internet.
Type of contract: Large-scale integrating project (IP)
contract No: 288535
Project URL: http://confine-project.eu

Editor:	Bart Braem, iMinds
Deliverable nature:	Report (R)
Dissemination level:	Public (PU)
Contractual Delivery Date:	15/11/2013
Actual Delivery Date	20/09/2013
Suggested Readers:	Project partners, future open call partners
Number of pages:	32
Keywords:	WP4, testbed design, requirement analysis, experimental research, community networks, testbed
Authors:	Bart Braem, iMinds Johan Bergs, iMinds Christoph Barz, Fraunhofer FKIE Jonathan Kirchhoff, Fraunhofer FKIE Henning Rogge, Fraunhofer FKIE Leandro Navarro, UPC Georgios Paitaris, AWMN
Peer review:	Ester Lopez, UPC Blaine Tatum, OPLAN

Abstract

This deliverable is the logical continuation of deliverable D4.1 in year 1, and presents ongoing research from the CONFINE project.

The first chapter faces on the typical challenges from working with community networks: their large scale, their heterogeneity and the limited resources available. Studies on both Guifi.net and community networks around the world are presented, followed by research on OLSR to overcome these challenges. Synergy effects were realized by building on the concepts and implementations realized in year 1. For example, the code basis used for implementing the radio to router communication protocol was also used for the implementation of OLSRv2.

The second chapter focuses on cross-layer interaction as a solution to enhance network performance. Both metric improvements and a more general cross-layer analysis are discussed in detail. The modular design of OLSRv2 allows for an integration of new metrics without breaking the compliance to the standard. For example, a new metric IETF draft was submitted to the IETF as an add-on to the standard metric to be able to cope with technology heterogeneous radio links.

The third and final chapter gives five studies on self-management in and around Community-Lab, to improve the overall success of the testbed. The results achieved are not only relevant for the testbed but for the operation of community networks in general.

Contents

List of Figures

List of Tables

1 Scale, heterogeneity and limited resources in the infrastructure

During the second year of the CONFINE project, more research has been focused on the study of the scale, heterogeneity and limited resources of community networks in general and the networks involved in CONFINE in particular.

In this chapter the project presents more insights in the high complexity of community networks, followed by a number solutions to tackle the challenges which come with this complexity.

In what follows, the first section gives an overview of studies performed on Guifi.net, which all focus on the complexity of the network. Next, results from a survey of community networks around the world are presented, which again highlights the complexity of community networks. Sections 1.3, 1.4 and 1.5 introduce improvements to the Optimized Link-State Routing protocol (OLSR) to cope with this complexity. E. g. Heterogeneity is addressed by the modular structure of OLSRv2 which allows for a simple integration of new routing metrics and the adaption of the Expected Transmission Time (ETT) metric to the directional link concept of OLSRv2. The ETT addresses radio links with different data rates. Finally, sections 1.6 and 1.7 present a monitoring solution and an antenna improvement developed in a community network.

1.1 Guifi Studies

Several studies have been performed on aspects of community networks related to its scale, heterogeneity and limited resources. The studies focus on the characteristics of Guifi.net or regions within it. It's our intent to generalise the studies covering other community networks.

1.1.1 Measurement and Analysis of a large-scale Wireless Community Network

Our first research topic corresponds to an ongoing research work not yet published on an extensive study of the characteristics of Guifi.net, the largest community network in number of nodes[1].

We present a measurement study of Guifi.net, a free, neutral and open access wireless community network consisting of more than 20,000 operational nodes. Guifi.net is the world's largest community network by the number of nodes and its coverage area. We used open data published by Guifi.net about the network nodes and wireless links, monitoring information, community database and web, and social data from the open community mailing lists and web portal. The time period of the data covers all 7 years of lifetime of the community. The scale and diversity of the network has sometimes required analysing subsets of the whole data separately. However, this has helped to identify local characteristics caused by demographic and geographic factors.

Our study looks into five subtopics: i) computer network diversity, ii) topology characteristics, iii) analysis of robustness, iv) social network analysis of the community structure, growth, and social participation, v) analysis of user experience. We analyse how the social community, the technology, the region where the network is deployed, and its self-organised structure shapes up the community network, and determines its properties, strengths and weaknesses.

A common community membership license or agreement defines the rules for participation in a free, neutral and open access network with a very active social participation in the construction and operation of the network, on its governance and its social life. Under a common license we find a unexpectedly large diversity of technical choices in hardware, software, routing and application protocols. Diversity brings complexity but also provides flexibility and strength to collectively experiment, learn and take advantage of alternatives. The results support the network freedom and neutrality claim.

The macroscopic topology structure is the consequence of an organic growth where new locations, links, and services are deployed under an immediate need. However, some areas show a perfect adaptation to their geographic and demographic characteristics that suggest some topological planning. In rural areas more long distance links are to quickly cover their area, while in urban areas mesh technologies are used due to shorter distances and more obstacles and interference to longer distance links. As a result, we can classify the Guifi.net network as a decentralised planning network, driven by "bottom-up" global and local growth.

The results of studying the hardware heterogeneity and configuration diversity of the network support the network freedom and neutrality claim. It makes it easier to introduce new users, but also threatens the stability of the network. We have found social patterns that describe how the network planning and growth is affected by users' behaviour i.e. network expansion over the time or locality impact of top active users actions. The network is robust to random failures (availability and reachability of network nodes) which are quite common, but not so robust to attacks.

Different profiles of community participants have diverse but complementary contributions. While some focus more on the network expansion and operation, others focus more on participation in the social part such as discussions and governance of the network, documentation, social events, support, training, etc. There are also cases contributing in both areas and a few members are remarkably and surprisingly active in the network, like human hubs or leaders for the different activities. The network has been growing continuously but we see a typical pattern of growth in extension and then density, where long links are deployed first that enable and promote a later growth in shorter links in new geographic areas.

We find high density Scale-Free network topology. However, we also find that the country's geographic and demographic heterogeneity, induced the creation of sub-network clusters with other topology patterns. The robustness analysis shows that network has critical points of failure. In general, nodes whose disconnection will provoke network partitions.

The user experience, exemplified by web proxy access shows that network proximity or other technical criteria may not be the determining reason for clients in selecting a proxy server, but the social relations and reputation among proxy owners have a key role. Finally, the evolution of the network, the community and technology suggests several directions of work, that are being explored by the community and in cooperation with the research community via this and other related projects.

Several opportunities emerge in the form of faster radio and fibre links in a falling market, which help to increase the capacity of the network. Lower cost long distance fibre links implemented as community fibre deployments or inexpensive dark fibre operators allow to increase the capacity, robustness and to reduce the network diameter inherent in pure wireless networks.

Innovations regarding routing protocols, particularly mesh routing protocols, combined with more powerful hardware, allow more self-configured and self-managed networks. This is particularly useful to lower the barrier of entry for new nodes deployed by new participants and to take advantage of urban environments full of interference and physical obstacles.

Inexpensive, water proof devices with Wi-Fi cards and directly attached wireless antennas with bridg-

ing capability connected by Ethernet to a central router allow the construction of modular and scalable nodes with multiple radio interfaces without having long antenna cables with their inherent attenuation (hybrid nodes in Guifi.net terms). This development was enabled by the CONFINE contribution to radio to router communication protocols[2] and their implementation (see CONFINE deliverable D4.1).

Community networks are much better supported by open source software tools developed by and for communities, covering all aspects from social networking tools, network management, network traffic, hardware drivers, network applications and services, combined with the GNU/Linux ecosystem of tools for networks. These tools have also derived in de-facto standards for the interoperability of the community networks.

Upper layer protocols, such as service discovery protocols allow network services to take advantage of network locality (one particular case of cross-layer interaction), critical for the idea of community networks but also to optimise the usage and performance of such networks.

Beyond that, the contribution of more powerful computers with more computing and storage capacity has enabled community clouds: platform services providing decentralised resource allocation of virtual nodes, networks and storage; platform services offering structured storage, computation, coordination, and communication services; and application services that can be deployed on demand to the community cloud and adapt to the elastic demand of its decentralised users.

1.1.2 Experimental Evaluation of a Wireless Community Mesh Network

The second research topic corresponds to a study on an area inside Guifi.net that uses the same routing protocol and forms a mesh network[3], which will be presented in November in an ACM conference.

Nowadays there are inexpensive Wi-Fi devices that have fostered the deployment of wireless communities. Well known routing protocols used in the Internet do not fit well to the characteristics of wireless networks. This has motivated an intensive research on routing protocols for wireless mesh networks. At this time, there are a number of mature and stable implementations that are being deployed in production networks. In this paper we focus on the experimental evaluation of a production Wireless Mesh network being deployed in a testbed at UPC and a quarter of the city of Barcelona, Spain, deployed using Quick Mesh Project (qMp) a firmware for embedded network devices based on OpenWRT Linux operative system. To our best knowledge, this is the first paper where a production community wireless network using 802.11an is analyzed.

From the study, we were able to draw a number of conclusions.

qMp Sants-UPC (QMPSU) is rather dynamic due to many reasons, e.g.: its community nature in an urban area; it is a growing network; there is a high diversity of the quality of wireless links; the mesh nature of the network. Characterizing such a dynamic network is challenging. To do so we have performed an extensive statistical study of the main parameters. These include topological properties, Internet access, usage of the network and characterization of the radio links. We have found simple distributions that fit some of these parameters. For instance, the network is not scale-free, the link length and traffic is fitted by a mixture of two exponentials, and the average throughput of the links is exponentially distributed. Regarding radio links, we have observed a non negligible asymmetry. Our results show that the network is rather well connected and adaptive. Thus, demonstrating the advantages of a wireless mesh network. Furthermore, even if the network is deployed in an urban area with an average link length of around 500 m, an average link throughput of around 14 Mbps was obtained. This high performance can be attributed to the 802.11an devices used in the network.

1.1.3 Software Defined Networking for Community Network Testbeds

The third research topic corresponds to a study on how Software Defined Networking can help to address the need for isolation and control of the network, virtualisation in other words. This study will be presented in October in an IEEE workshop associated to the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)[4].

Wireless Community Networks have received increasing attention the latest years. In an effort to set the cornerstone for an internet without central authorities and monopolies, network engineers throughout the world have started creating community networks. To enhance this effort, Community-lab, a wireless community networks testbed was created, allowing researchers to experiment with new protocols and applications in a realistic environment. Nevertheless, this testbed does not offer the ability to perform link layer experiments. To address this gap, we developed a platform that allows Community-Lab researchers to modify the link layer connectivity of the network and thus to perform certain types of link layer experiments. Moreover, we decided to reach our goal using Software Defined Networking (SDN) techniques, due to the attention they received lately and their promise for a complete networking solution. Overall, we propose an architecture that allows researchers to perform link layer experiments in a generic community network (CN) testbed. To prove the feasibility of our architecture we implemented it for Community-Lab using the OpenFlow SDN protocol[5], enabling researchers to manage the link layer topology of their experiment.

From the study, we concluded that is feasible to create a system that allows link layer experiments in CN testbeds using SDN techniques that provide virtualisation and isolation adapted to the characteristics of community networks. We proposed a generic architecture that fulfils our goal and implemented this architecture for an existing CN testbed, Community-Lab. Our implementation consists of two, Free and Open Source Software (FOSS) licensed, software components: Poxy[6], a proxy for OpenFlow traffic, and Pongo[7], an integration of POX OpenFlow controller, Community-Lab testbed server and Django. Using these software components we deployed the proposed architecture in Community-Lab, creating an application that allows managing the L2 topology of a researcher's nodes. Thus, we proved the feasibility of our architecture and the proper function of our implementation. Moreover, we performed a performance analysis of our system reaching the conclusion that the wireless multi-hop environment and the use of L2 mesh routing protocol are the greatest cause of overhead. Finally we discussed how our effort satisfies our goals and under which conditions our architecture could be used in a generic community network or wireless mesh network environment.

1.2 A Survey of Community Networks

As part of the analysis for this task, we set up a questionnaire for community networks around the world to learn about their infrastructure. Besides gathering knowledge about community networks, this questionnaire also serves as a reference for researchers about the characteristics and as a comparison source for community networks. E.g., when a researcher wants to simulate a community network, he or she can learn about typically used protocols or network scale, while a community network can learn from solutions other community networks have developed in the past. Also, this is a good opportunity to collect meta data which can be published as open data sets. More information on the CONFINE open data efforts can be found in deliverable D4.8.

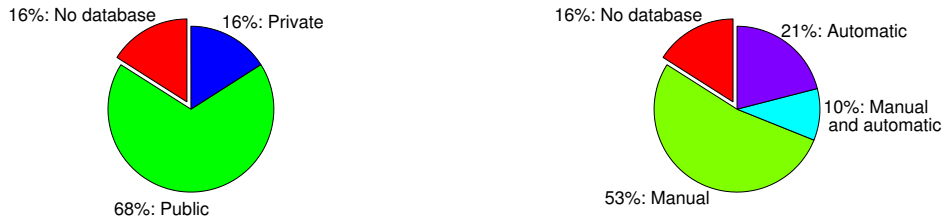
The questionnaire focused on both technical and non-technical aspects, but in this deliverable we will only focus on the technical aspects. The complete results will be presented during the International Summit for Community Wireless Networks 2013 in Berlin, and in a paper to be presented during

the International Workshop on Community Networks and Bottom-up-Broadband (CNBuB) '13 in Lyon[8]. It will be clear from the results presented below that heterogeneity is a serious challenge to community networks.

Node Database

We asked how the infrastructure of the community network is documented. Three (16%) community networks do not use any kind of database to keep track of their nodes and links in the network. Of the sixteen (84%) that do have some kind of database, thirteen (81%) allow public access, often via a web interface, to this node database. For the other three (19%) this database is held private. These results are depicted in Figure 1.1(a).

.



(a) Access policy . (b) Updating method

Figure 1.1: Statistics about the community network infrastructure database.

Updating the database of the community network, such as adding nodes and updating link information, is done exclusive manually by ten (62%) of the sixteen community networks community networks that have a node database. For four or 25% of those community networks, the database is updated automatically. A combination of manual input with automatic updates of the links is used in two of those community networks (12%). These results are shown in Figure 1.1(b).

Nodes

The number of nodes in the network is unknown for one community network. For sixteen (89%) of the remaining eighteen community networks, the number of nodes ranges between 5 to 500 with the mean around 84. Two community networks have more than 1000 nodes with one more than 10000 nodes. The distribution of the number of nodes on a log scale is depicted in Figure 1.2.

Links

For five community networks the number of links between the nodes is unknown. Of the remaining fourteen, eleven (79%) have a number of links ranging from 2 to 300 with the mean around 67. Three community networks have a number of links exceeding 1000 with one more than 10000 links. Figure 1.3 depicts the distribution of the number of links on a log scale.

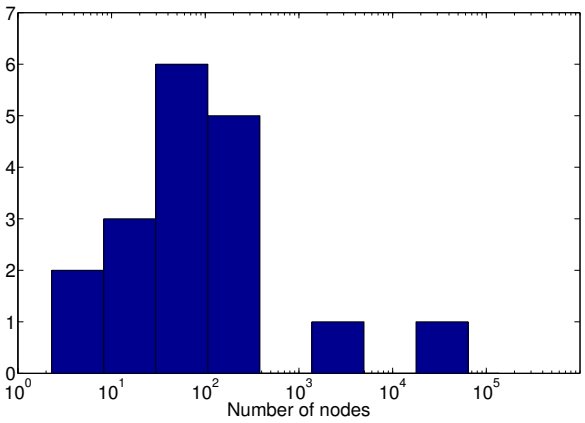


Figure 1.2: Histogram of the number of nodes in the community networks.

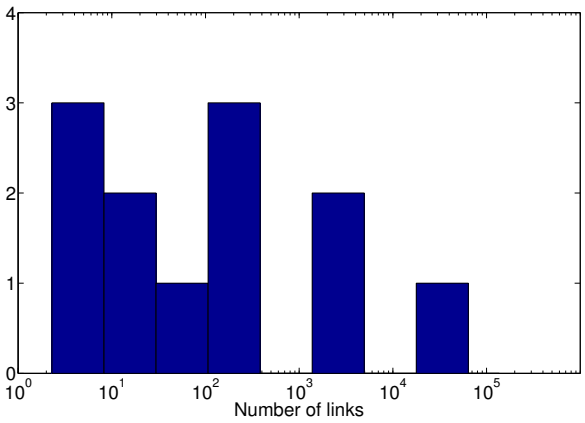


Figure 1.3: Histogram of the number of links in the community networks.

Hardware

The different community networks often use a specific hardware configuration. We provide the top three of hardware used. For one community network the answer was unclear and is, for this question, left out in the statistics. The following top three is based on eighteen community networks. In first place, used in ten (56%) community networks, is the *Ubiquiti*[9] hardware. In second and third place are *MikroTik*[10] and *Atheros*[11] both mentioned in three (17%) community networks. All three community networks that use *MikroTik* hardware also mentioned using *Ubiquiti* hardware. And *Ubiquiti* hardware is also used in one community network of the group that uses *Atheros*.

Figure 1.4 depicts these results. All nineteen community networks use some type of wireless tech-

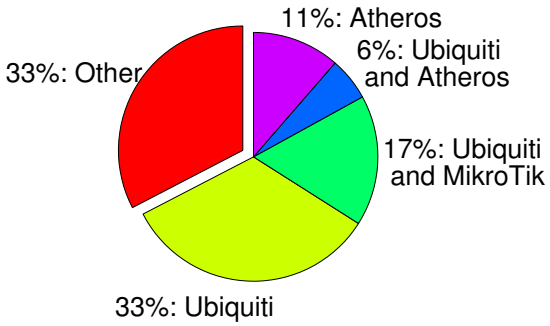


Figure 1.4: Hardware used in the community networks.

nology complemented with optical fiber, xDSL or mobile infrastructure. The 802.11n technology is used in nine (47%) community networks.

We take a closer look at the wireless hardware vendors used in the community networks. The following results are based on the fifteen community networks that answered this question. The top consists of, in first place, *Ubiquiti* used by eight (53%) community networks and in second place, *TP-link* used in three (20%) networks. In two (13%) community networks both *TP-link* and *Ubiquiti* hardware are used. The remaining six (40%) use a variety of other hardware vendors.

Software

For sixteen (84%) community networks, the preferred operating system used on the hardware is *OpenWrt*[12]. Other linux distros are mentioned in five (26%) community networks. In three (16%) community networks *RouterOS*[13] used by *MikroTik,* and also in three (16%) *AirOS*[14] used by *Ubiquiti,* is mentioned. *RouterOS* and *OpenWrt* are both mentioned in five (26%) community network's answers. These results are shown in Figure 1.5.

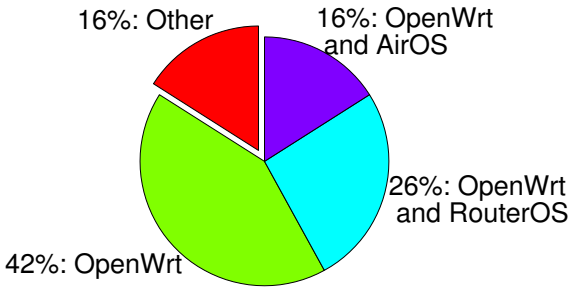


Figure 1.5: Operating systems used in the community networks.

We had a more detailed look at the software, protocol, or algorithms used for the routing in the community networks. In first place, used by ten (53%) community networks, comes *OLSR[15]* as the main routing protocol. Used in three (16%) community networks and in second place is *BGP*[16]. Third place and used in two (11%) community network, is *BATMAN*[17]. The remaining four (21%) use some different type of routing protocol. Figure 1.6 depicts these results.

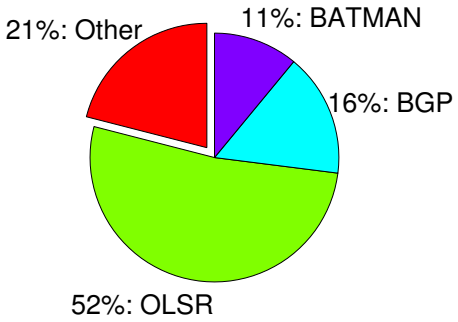


Figure 1.6: Routing protocols used in the community networks.

User access

In eight (42%) community networks there is no separation between the end user's access network and the backbone network interconnecting the end user's access network. In the remaining eleven (58%) community networks, the end user is separated from the backbone network.

1.3 Multi-topology extensions for OLSRv2

Not all traffic on a network is equal. While most community networks don't integrate the concept of priority traffic into their networking mechanisms to retain fairness among users, there are reasons why some traffic could be prioritized or restricted to a subset of links.

Certain real-time applications like voice over IP or video communication need reliable low-delay links. On the other hand, mass downloads like Bittorrent are more robust against delay and packet loss. To allow a Community Mesh Network to apply different metrics and linksets to traffic, we have implemented at FKIE a Multi-Topology extension for our OLSRv2 implementation. With this implementation it is possible to run multiple link metrics on the same routing protocol control messages and create several routing tables on each router.

Just before the IETF meeting in Berlin we learned that one of the authors of OLSRv2, Christopher Dearlove, is also working on standardizing and implementing a Multi-Topology extension for OLSRv2. After a long email exchange and very interesting discussion at the IETF in Berlin both sides recognized that the current designs of the extension are incomplete.

Both sides are confident that we identified the difficult parts of a full Multi-Topology extension together in Berlin and plan to work together on a coming standard. We have also talked about doing an interoperability test as soon as a new draft document about Multi-Topology and two matching implementations are available.

1.4 OLSRv2 Message Format Efficiency

One of the advantages of the OLSRv2 protocol compared to its parent RFC 3626 (OLSR v1) is the introduction of a type-length-value based message format. The new message format makes it much easier to extend and improve the existing protocol without breaking compatibility with existing implementations.

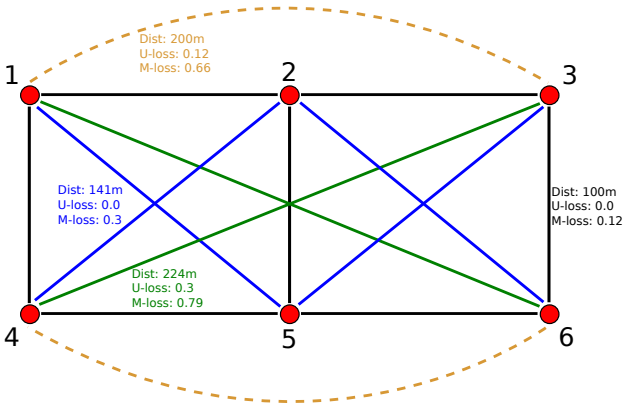
While the new message format is very flexible, it pays for this flexibility by adding overhead in the form of type and length fields. This overhead is partly compensated by the packet format address compression, but was not clear how these mechanisms compare to the overhead added by the TLVs.

Thus, we started a series of emulations to compare the control plane overhead of OLSRv1 and OLSRv2. The emulation was done as virtual machines running a real implementation of both routing protocols with comparable timings. The overhead was then measured for each of the protocols with different topologies connecting the virtual machines.

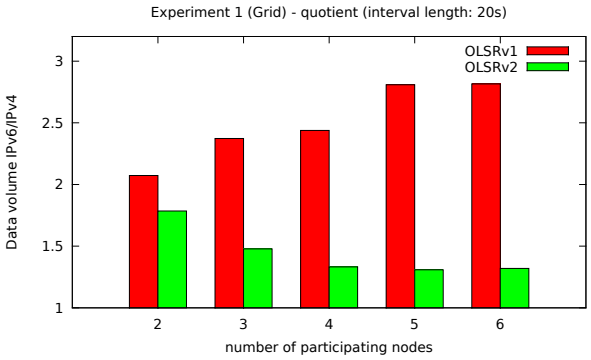
In the topology depicted in Figure 1.7(a), six nodes are placed on a 2×3 grid with 100 m distance between direct neighbors. Due to the short distance between the nodes and the size of the grid, all nodes are connected via a direct link. This results in 5 links per node. Loss rates are calculated from the Cartesian distances between each pair, which results in practically no packet loss between direct neighbors and high packet loss between opposing nodes.

The aim of the experiment was to compare the overhead of OLSRv1 and OLSRv2, in particular with regard to the effects of IPv6 address compression on the protocol overhead. The address compression of OLSRv2 generally delivers the better compression ratios the larger the network mask of the IPv6 address. Data volume has been measured on a bridge by capturing all packets with libpcap. Beginning with a steady-state situation, additional nodes were gradually added to the network.

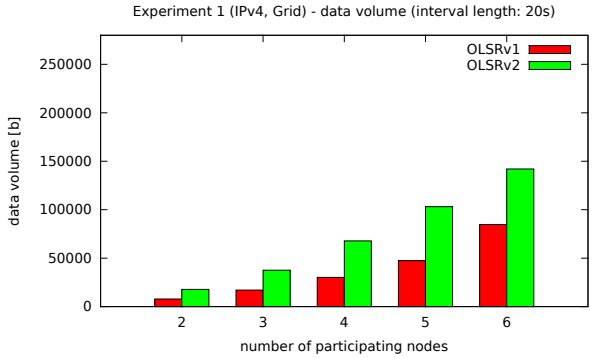
Figure 1.7(c) and 1.7(d) show the results of the experiment for the grid topology. Figure 1.7(c) shows the total amount of traffic in bit sent by all nodes for OLSRv1 and OLSRv2 in an IPv4 only



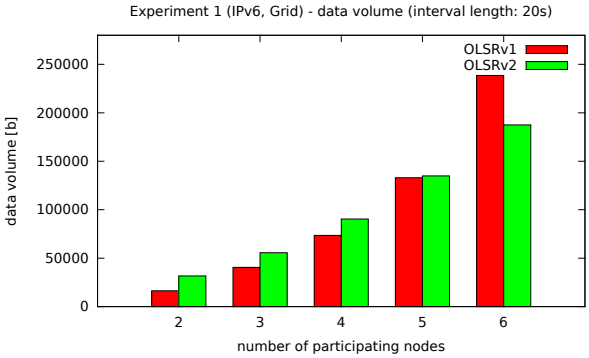
(a) Experiment topology



(b) IPv6 overhead compared to IPv4



(c) Total amount of data sent - IPv4 only



(d) Total amount of data sent - IPv6 only

Figure 1.7: Overhead comparision of OLSRv1 and OLSRv2 with IPv4 and IPv6

configuration. Figure 1.7(d) shows the same for IPv6. Please note that the total amount of traffic using an IPv6 only configuration is larger than using an IPv4 only configuration for both OLSRv1 and OLSRv2. Regarding IPv4 the traffic generated by OLSRv1 is significantly lower than using OLSRv2. This is due to the TLV based packet format of OLSRv2 which, while being more flexible in terms of extensibility, also comes with a higher overhead than the binary packet format of OLSRv1. However, when using OLSRv2 in an IPv6 only mode, the address compression of OLSRv2 manages to keep the amount of traffic generated well below the values for OLSRv1 for a larger number of nodes. For 5 nodes, the break-even is reached. For 6 and more nodes, OLSRv2 provides better results. Please note, that we carefully selected the addressing scheme for the compression mechanism to be most efficient.

Figure 1.7(b) provides another view on the results, as it shows the percentage of traffic generated by IPv6 in comparison to the IPv4 only configuration for both protocols. One can easily notice that the overhead of IPv6 compared to IPv4 is much higher for OLSRv1 than for OLSRv2. In addition, with an increasing number of nodes, the ratio seems to converge. However, this has to be verified in future experiments with a larger number of nodes.

More details and further experiments have been published on the The 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2013) [18].

More measurements in larger emulation scenarios and in the CONFINE testbeds will be necessary to confirm and extend the results in the coming year.

1.5 Routing Filter Implementation for OLSRv2

Distributing the routes of a routing protocol on the routing tables of the underlying operation system can be very deployment specific. While the OLSRv2 implementation of CONFINE can configure a custom routing table for each topology, the core doesn't allow the user to set a more specific set of rules.

One common use case for such a configuration is to put the default route into a special routing table. Such a split makes it easier to set up Network Address Translation (NAT) rules for a router. Other Community Mesh Networks have similar or different configurations.

To allow a user to setup a more complex distribution of the routes without overloading the core of the OLSRv2 implementation with too many options, a Routing Filter API has been implemented.

This API allows plug-ins to hook into the control path between the routing calculation by the Dijkstra Algorithm and the code that sends the routes to the operating system. The example implementation allows the user to overwrite routing table, protocol and distance for any IP prefix.

While this plug-in should be useful for community mesh networks, the Routing Filter API also allows the plug-in to prevent the core from setting up a route at all. This feature should allow hooking the OLSRv2 implementation into the Quagga Routing framework in the future without Quagga specific code in the core.

1.6 A monitor for community wireless networks

This section discusses a monitor for wireless community networks that use BGP as their routing protocol. The monitor has access to the node database, BGP daemons, routers, and sensors across the network and notifies problems to number and naming authorities, and node operators. It also draws a near real time connectivity graph and can automatically trigger actions on events.

1.6.1 Background and the AWMN Model

In community networks, people own, setup, and maintain network and radio equipment on a best effort basis. Most wireless nodes in the Athens Wireless Metropolitan Network (AWMN) are configured as autonomous systems (AS) and the vast majority of inter-AS links are wireless. The networking and radio equipment is exposed to harsh environments and equipment failures, configuration errors, equipment incompatibilities, and manipulation of the infrastructure are more common than in professional ISP networks. Such community networks have number and naming authorities (host-masters). However, there is not a single authority that has administrative access to all the networking equipment.

Usually there are node databases yet they are often incomplete and contain a large amount of false information. There are no peering agreements and established protocols to deal with problems and community spirit and good interpersonal relationships between the node operators are not always enough to cope with these problems. Also the vast coverage areas and the large number of network nodes add to the complexity of community networks.

1.6.2 Definitions

1.6.2.1 Node Database

A node database is a database with IP addresses, node geographical coordinates (latitude, longitude, elevation), node operator contact information, DNS zones, link information and information for services provided by various nodes across the network. The number and names authorities of the network allocate and assign resources via the node database and most information is added to the database by the node operators. For example, in the AWMN, the Wireless Node Database (WiND[19, 20]) is used.

1.6.2.2 Node

A node is an autonomous system as defined in RFC 4271 where the most inter-AS links are wireless and most links within the autonomous system are wired.

1.6.2.3 Node Operator

A node operator is a community member who acts as a node administrator to one or more nodes. This member usually has a wireless node on the rooftop of his residence.

1.6.3 Border Gateway Protocol Monitor

Our proposed network monitoring system has access to many BGP speakers across the network. In a non-professional, non-homogeneous wireless network, AS-paths change very often and it is very common to find wrong AS-paths, ghost links and ghost prefixes in a BGP table. Although we have access to many BGP tables, we need a way to attach levels of certainty to the information they contain.

Also in community networks, configuration mistakes and pranks such as IP hijacks are common. The following procedure attempts to deal with ghost links, ghost prefixes, configuration mistakes, and IP hijacks by using the information on BGP tables from across the network.

1. The BGP tables are fetched from the network in short regular intervals, e.g. 30 minutes.
2. BGP prepends are detected and filtered out the data.
3. A path that contains an AS multiple times contains a loop. This loop should be notified and the path is excluded for the following steps.
4. The BGP table is split in pairs of AS numbers that form a links.
5. A level of certainty is assigned to each pair. The further away the discovered link is in hop-count, the lower level of certainty it gets. This weight for a pair is calculated by adding a value of one for every hop starting from the node for which the table is analysed. A higher weight value means a lower certainty level. For example, for the AS-path $2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ on the node with AS 1, the weight of the pair $(1, 2)$ is $0 + 1 = 1$, the weight for the pair $(2, 3)$ is $1 + 2 = 3$ and so on. Because the pair $(1, 2)$ is closer to AS 1 than pair $(2, 3)$, pair $(1, 2)$ is assigned a higher level of certainty and thus a lower weight.
6. While traversing the BGP tables of the network, the weight of a pair is only replaced with lower weight values. This results in a list of pairs with their highest level of certainty.
7. Invalid links should be notified. For example, if the weight of a pair (x, y) for AS x equals to 1; meaning that the BGP table of AS 1 is analysed. A weight for $(x, z) > 1$ means that another

AS has the pair (x, z) as a link in a path yet AS 1 has no knowledge of this link. In this case the pair (x, z) should be triggered as invalid.

8. For every prefix in the BGP tables, the assignment of this prefix by the number authority is validated. This means verifying it exists and it is assigned only once. By using the information of the number authority, the malicious prefixes can be located and triggered to the host-masters.
9. If a path for a prefix leads to a node-AS for which a BGP table is available and the node does not announces this prefix then this prefix is triggered as a ghost prefix.
10. For paths for a prefix to a node-AS for which no BGP table is available, the weight of the level of certainty is used to guess ghost prefixes. Similar as for the link pairs, the weight of the prefixes is replaced only if a lower weight value is found and thus indicating a higher level of certainty.

1.6.4 Additional information to monitor

The monitor should be complemented with information from monitoring systems such as Nagios[21]. Also trace paths between AS nodes can provide additional information in discovering asymmetric paths.

Because it is a wireless network that is being monitored, values such as Client Connection Quality (CCQ), signal/noise ratios,noise level,and signal strength should also be monitored. Such values are often easily retrieved from the nodes via SNMP[22] and can be added in a monitoring system.

1.6.5 Intelligent Routers

Most recent wireless routers are fully embedded systems with a large amount of CPU power and RAM. By running additional scripts on the routers, the nodes can exert discovery and healing capabilities.

For example, it is not uncommon that dish antennas are misaligned by strong winds. The quality of the link via the dish antenna may drop drastically yet the BGP session will remain active and packets are still routed over this link. A script[1] could monitor the signal strength of a link and close the appropriate BGP session when the script detects a significant drop in the signal strength.

1.7 Experiments with 3x3 MIMO antennas

1.7.1 Introduction

In AWMN, offset dish antennas with 5GHz copper feeders are used for most long range backbone links. These offset dish antennas are low cost, have relatively high gain and cope well with the Greek weather. The unlicensed parts of the spectrum that the members of the AWMN may use freely in Greece, are 2.401GHz-2.484GHz and 5.475GHz-5.725GHz. In large cities, like Athens, where thousands of backbone links are operating close to each other, spectrum conservation becomes important. Any wireless protocol, RF (Radio Frequency) technology, antenna or guidelines that can conserve spectrum are welcomed by the AWMN members.

[1] Example script by SV1BDS:
http://alog.ipduh.com/2013/09/enable-disable-bgp-peer-based-on-signal.html

The 2x2 Multiple In Multiple Out (MIMO) in 802.11n and Time Devision Multiple Access (TDMA) protocols have been a success in the AWMN. With the 2x2 MIMO, one channel and two polarisations are used to transmit two streams and conserve spectrum usage. This way, more bandwidth can be accommodated by the network with less interferences and less equipment. Recently, 3x3 MIMO cards are available for low prices. AWMN decided to construct 3x3 MIMO feeders that can be used with an offset dish in long range links.

This section discusses 3x3 MIMO long range antennas and describes the tests preformed and planned in the quest to construct practical 3x3 MIMO long range antennas. One of the major goals is to achieve data rates above MCS (Modulation Coding Scheme) 20 index.

1.7.2 Methodology

The methodology was straight forward. First, a lot of time and effort was invested in researching and collaborating. Then, some initial tests and visualisations of the radiation patterns of the feeder designs were set-up with NEC software. At last, field tests and experiments were conducted and evaluated.

1.7.3 Challenges with 3x3 MIMO

Crossing the gap between 3x3 MIMO with 2 spatial streams and 3x3 MIMO with 3 spatial streams was challenging because we did not want to use four or six dish antennas for each point to point link. Neither did we want to use many feeders or set up huge reflectors. The ideal setup would be two dish antennas and 2 feeders per point to point link. The parameters that were adjusted to differentiate the data spatial steams are polarisation, distance between monopoles and LMR cable length.

1.7.4 The 3x3 MIMO feeders

So far, three types of feeders are constructed. We used the classic AWMN feeder that works well with offset parabolic dish antennas as a base. Then we placed the monopoles in various positions on the x and the y-axis to achieve more polarisation and better isolation in the case of 0°, 90° between polarisations.

We constructed the following feeders.

1. Monopoles in 0°, 90°, 225° all on the same coplanar level.
2. Monopoles in 0°, 120°, 240° all on the same level.
3. Monopoles in 0°, 90°, 270°, the 0° and 90° monopoles and their reflectors are on a different level.

1.7.5 Tests

We define a test as passed when the following requirements are met:

1. Three spatial streams are used.
2. MCS index is 20 or higher.
3. The UDP throughput over the link exceeds 220Mb/s over a 40 MHz channel.

1.7.5.0.1 Test 1 This test is performed indoors. There is a spacing from a few centimetres up to 10 meters between the nodes. Each node has three small $\lambda/4$ antennas on each node. For the experiments ranging from few centimetres up to 2 meters, the results were excellent. We saw the wireless registration locking at 450Mb/s and bandwidth tests reporting more than 300Mb/s UDP traffic. This test passed the requirements.

1.7.5.0.2 Test 2 and 3 This test is performed indoors. The distance between the nodes was 5 and 10 metres. Each node has one 0°, 120°, 240° feeder. We saw momentarily rates above the MCS 20 index yet the test failed the requirements.

1.7.5.0.3 Test 4 This test is performed indoors. The distance between the nodes was 5 and 10 metres. Each node has one 0°, 90°, 270° feeder. The maximum stable data rate achieved was MCS 19, thus not fulfilling the requirements. This test failed.

1.7.5.0.4 Test 5 This test is performed indoors. The distance between the nodes was 5 and 10 metres. Each node had two feeders: one with horizontal and vertical polarisation, the other one with diagonal polarisation. This test passed.

1.7.5.0.5 Test 6 This test is performed outdoors. The distance between the nodes was 3.5 km. Each node had one offset parabolic dish antenna with a 0°, 90°, 225° feeder. We saw momentarily rates above the MCS 20 index yet the test failed the requirements.

1.7.5.0.6 Test 7 This test is performed outdoors. The distance between the nodes was 3.5 km. Each node had one offset parabolic dish antenna with a 0°, 90°, 270° feeder. A maximum MCS of index 19 was achieved, the test failed the requirements.

1.7.5.0.7 Test 8 This test is performed outdoors. The distance between the nodes was 3.5 km. Each node had one offset parabolic dish antenna with a 0°, 120°, 240° feeder. We saw momentarily rates above the MCS 20 index yet the test failed the requirements.

1.7.5.0.8 Test 9 This test is planned and will be performed outdoors. The distance between the nodes will be 3.5 km. Each node will have two offset parabolic dish antenna with two feeders: one with a vertical and horizontal polarisation and the other with a diagonal 45° polarisation.

1.7.5.0.9 Test 10 This test is planned and will be performed outdoor. The distance between the nodes will be 3.5 km. Each node will have one offset parabolic dish antenna with the 0°, 90°, 225° feeder and variable length of LMR cable. Tests are planned with 1 up to 12m LMR cables to affect the time on the third stream.

1.7.5.0.10 Test 11 This test is planned and will be performed outdoor. The distance between the nodes will range from 5 to 50m. Each node will omni antennas with feeders.

1.7.5.0.11 Test 12 This test is planned and will be performed in a large indoor space. Each node will use omni antennas with the constructed feeders.

The Tests were made by Georgios Paitaris and Nikolas Nikos aka nikolas_350. Many members of the AWMN community helped us. Special Thanks to George Katsimanglis aka SV1BDS and Joseph Boniciolli aka NetTraptor.

2 Cross-Layer Interactions and Optimizations

Cross-layer research in year 2 has focused on link quality metrics and ETT metrics on the one hand, and a cross-layer analysis approach on the other hand.

The metrics analysis should be considered a part of the general analysis, which should be performed cross-layer to include dependencies from the different layers.

The metrics research is a continuation of research performed during the first year of the project, while the cross-layer analysis strategy has been outlined because of research results of the first year which indicate a high network complexity.

2.1 Statistical Processing of Link Quality Metrics

OLSR.org is currently the most widely deployed mesh routing protocol in Community Mesh Networks. The protocol uses an Expected Transmission Count (ETX) metric to calculate the cost of each link for the Dijkstra calculation. While the implementation works, the ETX metric shows several problems that have to be addressed.

One problem is that most metric costs typically fluctuate, even on perfectly stable links. This is not a measurement problem but a result of the low amount of input data based on the packet loss of the OLSR control packets. Just by chance, the estimation of a link will get higher and lower by itself.

The current solution to this problem is calculating an average of the link cost over a longer period of time. This helps to smooth the link cost by including more measurements, but it also makes the metric slow to react to changing link properties.

We looked into statistical approaches to filter out the noise of a metric by using an exponential weighted moving average (EWMA) and variance. These two values can be calculated without storing the whole series of incoming packet loss events, which would make them easier to implement.

Unfortunately, EWMA plus variance still lack the necessary stability when processing binary input (packet loss). Even in mathematical simulations we were unable to find parameter sets which produced a reasonably smooth output without reacting too slow to changes of the links average packet loss.

We are now looking into different methods of post-processing the output of a metric to stabilize it and filter out the unwanted noise.

2.2 ETT metric draft and implementation

Based on the experience of the OLSR.org ETX metric implementation we added an Estimated Travel Time (ETT) metric to the OLSRv2 implementation. This metric takes both the directional packet loss and the link speed into account.

The implementation is just a variant of the ETT metric suggested in the paper “Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks”[23]. Instead of calculating the symmetric packet loss, it just measures the unidirectional packet loss and combines this with the link-speed. While this is a

slightly different approach compared to the ETT paper, it is a better fit to the directional link costs defined in the OLSRv2 protocol.

Our implementation currently supports three different sources of link speed. It can read the incoming link speed directly from the Mac80211 Wi-Fi stack of Linux, it can read the link speed of an Ethernet connection, or it allows the user to set a link speed in the configuration. Other sources for this link layer data, like the radio to router communication protocol DLEP, can be easily added.

CONFINE partners submitted an IETF draft [24] defining the new ETT metric. The draft has been presented on the IETF in Berlin and raised a lot of interest. The feedback from the IETF working group will now be integrated into the next draft release. At this point the MANET Working Group might decide to adopt the ETT draft as an official Working Group Document, which could become an Informal RFC of the IETF.

2.3 Cross-Layer Analysis of Community Networks

Open community networks provide interesting research features, as seen in [25, 26, 27], since they are often planned ad hoc and driven on demand. This is not only the case on the infrastructure level. Also the way they are used by the communities might reveal interesting patterns in space and time. To abstract from the complexity of those networks, we propose a number of analysis levels. By cross analysing those different levels, features of the original, complex network can be found and be used to improve it.

2.3.1 Analysis Levels

There are a number of levels as depicted in Fig. 2.1 in the network which provide an interesting

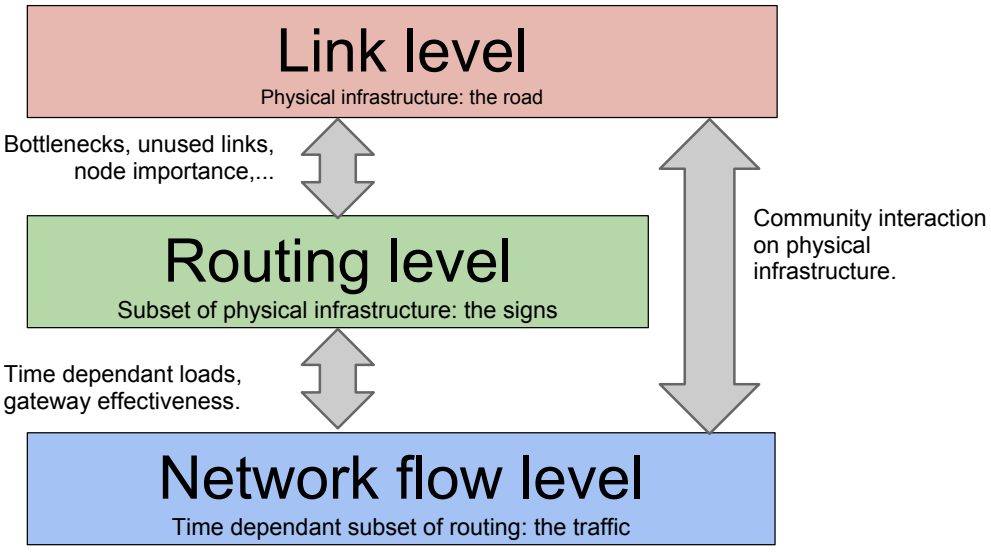


Figure 2.1: Analysis levels of community networks.

analysis. Some can be used to detect bottlenecks and provide insight into further network planning.

Others can provide information on the behaviour of the network in time, allowing dynamically altering the network conditions to better accommodate this behaviour.

2.3.1.1 Link Level (L2)

On this level, only the point to point connections between the network nodes and their channels are considered: the physical network topology. This topology can be abstracted as a graph with weights on the edges correlated to the capacity, delay, robustness, or other parameters. Using graph analysis algorithms such as max flow, min cut, Cheeger constant, those graphs can be evaluated regarding their structure and the distribution of the weight parameter to locate exceptional patterns like bottlenecks, underused links or interference. This level is independent on the routing protocols used in the network and thus provides insight into the theoretical capabilities and robustness of the deployed network.

2.3.1.2 Routing Level (L3)

Overlaid on the physical network topology, a routing graph is constructed by the routing protocol. Since this graph is a subgraph of the link level graph, the analysis of the routing level graph extends the analysis of the link level graph: for example, all bottlenecks in the link level graph will be bottlenecks in the routing graph. Yet, not all bottlenecks of the routing graph are link graph bottlenecks. In a similar approach like for the link level graph, exceptional features of the routing graph can be discovered. Different to the link level graph, the routing graph can vary in time depending on the routing protocol used. By comparing the routing graphs between two time intervals, interesting patterns of the routing protocols can be revealed.

2.3.1.3 Network Flow Level (L4)

While the link and routing levels are driven by algorithms and logic, network flows are mostly generated by humans and map the relationships between community nodes and the rest of the Internet. The network flows use the routing level to get to their destination. These flows can be found using tomography[28]. Analysing those flows can reveal patterns of communication between different communities both in space and time. They can be used to be correlated to certain real-life events or predict certain network flows. One of the challenges here is to capture this information in the network.

2.3.2 Cross Level Analysis

As shown in the previous section, each level has its own added value in the analysis. This information can be used in a cross level analysis, revealing cross level features of the network which in turn can be used to improve the network.

2.3.2.1 Link Level with Routing Level

Patterns found on the routing level can be mapped on the network infrastructure, revealing unused links. Those unused links can be used by the routing level to provide load balancing and fallback links. Also, the channel allocation on the link level could be altered to better accommodate the network routes laid down by the routing protocol. Physical links can be improved or made redundant when the network level analysis indicates them as bottlenecks even when the link level does not.

2.3.2.2 Routing Level with Network Flow Level

Two forms of network flows can be distinguished at the network flow level. One is the communication between nodes within the community network. Traffic between nodes found to be time related by the network flow analysis can be improved by the routing level to provide the necessary resources for this traffic. The other form is network flows destined outside the community network, which leave the community network at gateways. Using the analysis information provided at the network flow level, the placement of those gateways can be analysed and improved.

2.3.2.3 Link level with Network Flow Level

The network flows might show that two communities frequently communicate yet lack the necessary resources on the link level. This feedback can be used by the link level to prioritise new link installations or bottlenecks to be improved. Also research on network planning, such as [29, 30], can complement such cross layer research.

2.3.3 Preliminary Results

As an example we took the wireless community network Ninux [31]. We parsed the HTML information of the *Map* section. Starting from node *Ale-Nord*, 107 nodes were discovered in this connected network by following the links of each new node found. Using link specific parameters, weights are assigned for each link in the network. Because some links miss crucial information, the minimal weight is considered for them. Then, independent of routing and traffic information, the network is analysed on different parameters, for example, the loss experienced in the network. The result of such an analysis of a parameter is a 107×107 matrix. This matrix indicates in this case the amount of loss experienced between each pair of nodes, which can be seen depicted as a heat map in Fig. 2.2.

Each entry in the matrix is a normalised value between 0 (experiencing maximum loss of the network) and 1 (experiencing minimum loss).

Using this information, sub (indicated in red) or super normal (indicated in green) nodes with mean values respectively below or above one standard deviation of the average means can be indicated as shown on the network graph in figure 2.3.

The node names of those sub normal nodes are shown in table 2.1.

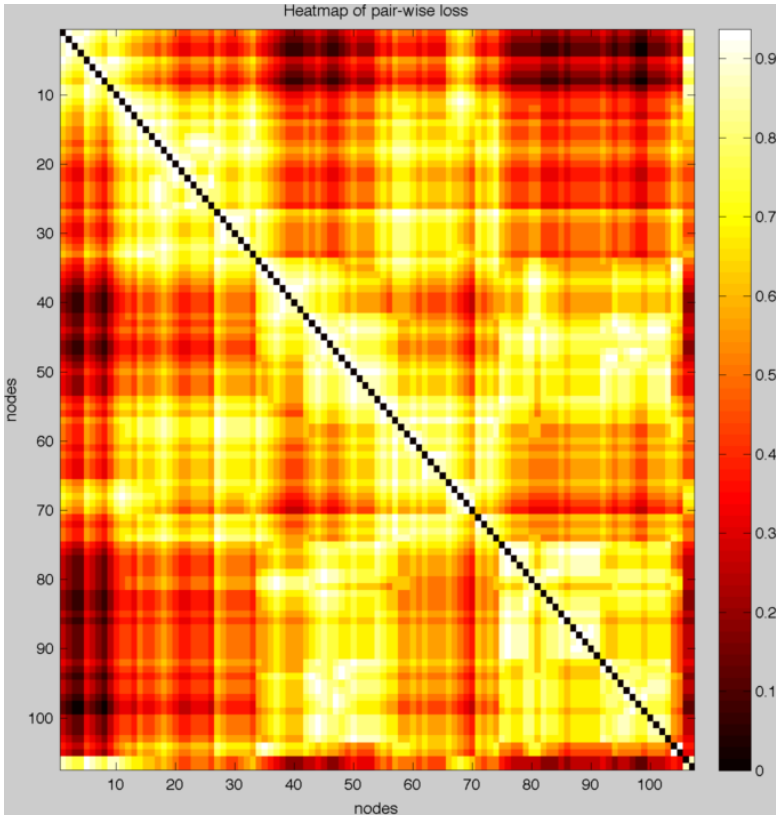


Figure 2.2: Heat map of the 107×107 node matrix

#	Name
8	Halnet-2
3	C5L
4	Morpheo
7	Halnet
2	Pachi
107	C5L2
9	Fabio
106	GrandeTony
6	NazzaNode2
1	Ale-Nord
70	emixHome
5	NazzaNode
99	Fiore Mar

Table 2.1: List of loss related sub normal nodes

This list indicates which nodes should be tackled when improving the network regarding the loss experienced by the nodes. The best way to improve the network is by connecting a red node with a green node. The green nodes have a low mean loss with all other nodes, resulting in the ideal candidate to connect to and lower the loss for the red node.

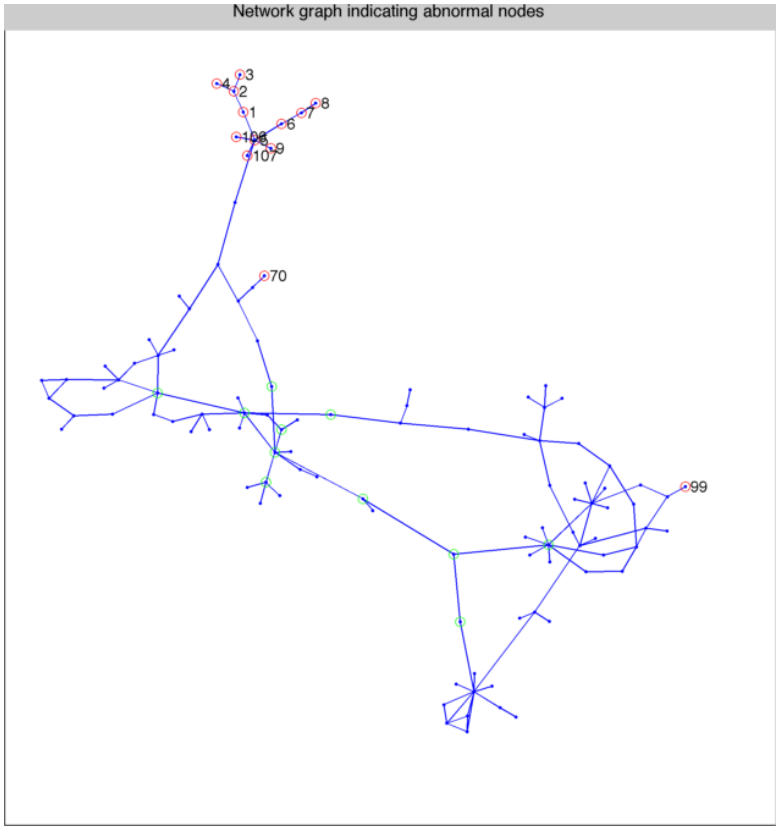


Figure 2.3: Network graph indicating abnormal nodes: red are sub normal, green are super normal

As mentioned before, some information about the links is missing. The more accurate the information from the network, the better the results of the evaluation and the more precise the conclusions will be. The following information highly improves the evaluation if available:

- The interfaces in the network.
- The nodes made up of multiple interfaces.
- The links between the interfaces.
- Mean and standard deviation of the link bit rate.
- Mean and standard deviation of the link SNR (Signal to noise ratio).
- Mean and standard deviation of the link load.
- Channels the links operate on.
- Routing table of each node.
- Routing metric for each routing entry.

If this information can be made easily accessible via an API, daily snapshots of the network can be evaluated and analysed.

3 Self-management

In this chapter, results from five different research works on self-management are presented.

Most studies focus on coordination based on environmental factors. This will lead to adapting the operation of a decentralized system with limited resources. In this case, the decentralized system is the CONFINE testbed or its environment, the community network. The resources are limited because of the characteristics of the community network itself.

The first two sections in this chapter focus on more general resource allocation strategies. Section 3.3 considers receiver-driven routing as a self-managing optimization strategy, section 3.4 describes a monitoring system essential for local and global optimization. Finally, section 3.5 introduces dual stack support for the OLSRv2 Neighborhood Discovery Protocol (NHDP). The latter is essential for link-local self-management.

3.1 Incentives for Dynamic and Energy-Aware Capacity Allocation for Multi-tenant Clusters

The first research work looks at the generic problem of resource allocation based on user requests while reducing energy costs by using mechanisms to dynamically scale up (buy) or down (sell) the allocated time slots of node and network resources[32].

Large scale clusters are now being used in shared, multi-tenant scenarios by heterogeneous applications with completely different requirements. In this scenario, it is useful to explore the intersection of two complementary goals. On one side, energy efficiency is an important factor to consider when being confronted with increasing operating costs related to energy consumption. On the other side, heterogeneous applications emphasize the problem of distributing the execution capacity among competitive users in a shared setting. Here, we address the combination of these two goals by introducing an incentive mechanism to make users report their actual resource requirements, allowing them to dynamically scale-up or down as necessary. In turn, this information is used by the infrastructure operator to shut down resources without reducing the Quality of Service (QoS) provided to users and effectively reducing energy costs. We show how our mechanism is able to meet the performance requirements of applications without over-provisioning physical resources, which in turn translates into energy savings.

Our mechanism pursues two different goals. From the user perspective, we show how our incentive mechanism effectively encourages users to report their true share requirement for a given job. Thus, we are able to provide a shares market in which users engage to dynamically scale up (buy) or down (sell) the allocated time slots of a job without the intervention of the infrastructure operator, providing a more elastic and agile infrastructure. We also show that it is in the users' best interest to participate in the market to improve their QoS instead of default. Our experimental scenarios can benefit from our mechanism as quality of service becomes more important than job run time as it affects experiment quality and repeatability.

From the infrastructure operator point of view, our mechanism is able to collect valuable information from users by providing them an incentive to truthfully report share requirements. In a scenario

in which energy related costs is one of the single largest factors in the overall cost of operating an infrastructure, this information can be used to shut down a portion of the resources without reducing the quality of service provided to users, which in turn could reduce energy costs. Looking at the results, we can conclude that our mechanism is a step towards a more rational use of the available resources. It is able to dynamically scale up and down the shares allocated to jobs with the aid of users and at the same time provide valuable information to the resource provider to shutdown spare resources without breaking Service Level Agreements (SLA) or affecting QoS.

To the best of our knowledge, our work is the first to explore the problem of the intersection of ensuring users' quality of service providing dynamic allocation of resources based on user requests and reducing energy costs together.

3.2 Effort-based Incentives for Resource Sharing in Collaborative Volunteer Applications

The second research work looks at incentives for resource contribution in collaborative volunteer applications where contributors are heterogeneous in amount of resources, looking at the percentage of contribution instead of just the amount of shared resources[33]. This can apply to any decentralised algorithm at any level of a distributed system, or particularly in resource sharing scenarios dealing with content, processing, or network link allocation (routing).

Collaborative and volunteer applications need to implement incentive mechanisms to regulate resource sharing and encourage network nodes to contribute for reaching a certain goal. Typically, these incentive mechanisms assign resources to network node requests, based on the total amount of resources contributed by the requesting participant. This approach assumes that participants contributing more should also get back more resources from the collaborative environment. This assumption turns the system unfair to those participants with scarce resources, because they have just few resources to share. This paper proposes the use of an incentive strategy based on the contribution percentage of each node, i.e. an effort-based approach. This proposal is evaluated and compared to contribution-based strategies. The obtained results show that the proposed effort-based approach not only benefits participants that have scarce resources, but also it is able to satisfy the requests of the powerful nodes.

The aim of this study is to understand different strategies for incentivizing a resource sharing ecosystem. We conducted a study based on simulations, which compared the impact of contribution-based and effort-based incentives on the resource sharing process in a scenario of hardware heterogeneity. Experiments on different scenarios show similar result trends regardless of the overlay network topology used. When an effort-based strategy was used, the nodes showed a higher willingness to cooperate, no matter how much resources they have. However when a contribution-based strategy was used, the nodes tended to cooperate only with their equals.

In global terms, the metric percentage of tasks fully satisfied is higher in the contribution-based strategy, because the nodes are self-providing the resources they need. This leads the system to have a lot of resources underused since the nodes do not have enough CPU to share with others. Hence, the resource sharing ecosystem becomes an inequality system where unique benefiters are powerful nodes that have enough resources to do their tasks and only occasionally collaborate with others. Collective satisfaction is achieved when most nodes are satisfied, although the percentage of success in the system is lower. It can only happen when all nodes can be equally graded like it occurs in the effort-based strategy. The differences between both strategies concern only participants with

scarce resources. Nevertheless, in effort-based strategies the cooperation relationship is fairer, giving powerless nodes the opportunity to fulfil their resources needs.

These results provide a first understanding of the impact produced by both strategies on cooperation, when nodes play the same strategy. However, in peer-to-peer and volunteer computing applications, the participants typically use different strategies with different nodes, depending on the parameter that they want to optimize and the current situation. The study of the impact of system dynamics is part of the future work.

Another point to address as future work is the neighbour nodes information inference. In every experiment we have assumed that all participants know the maximum amount of resources, i.e. CPU slots of the node that their neighbours can share at maximum. This is a strong assumption since it is not ease to obtain in real world applications. It is however a critical value needed to measure the effort of each node, when it has to decide with whom to share some slots. Therefore, our future work will be also focus on finding a way to identify the maximum resources available in each participant.

3.3 Receiver-Driven Routing for Community Mesh Networks

The next research work looks at how a routing algorithm can incorporate the preferences of receivers in routing traffic, and allowing to have routing algorithms that respect the community social contract and without restricting the freedom of community users[34].

Community wireless mesh networks are decentralized and cooperative structures with participation rules that define their freedom, openness and neutrality. The operation of these networks require routing algorithms that may impose additional unnecessary technical restrictions in the determination of routes that can restrict the freedom of community users. We propose a receiver-driven discretionary routing mechanism where each receiver (the intended destination of the packet) can freely specify delivery objectives and remain compatible with the collaborative approach of community networks. Each node has a unique identifier and can announce the description of its offer and also the description of its routing policy with preferences to deliver traffic to it. BATMAN eXperimental version 6 (BMX6), a wireless mesh routing protocol, provides a “hash-based profile propagation mechanism” to disseminate descriptions. This receiver-driven routing can be applied to express preferences for desirable nodes and paths, or to restrict traffic to trusted nodes enabling trust and security aware routing. We validate our contributions with a proof of concept implementation of key concepts, as an extension of the BMX6 routing protocol, that confirms its feasibility and scalability.

In this work we present a receiver-driven discretionary routing mechanism for community networks where each receiver can freely specify delivery objectives and remain compatible with the collaborative aim of community networks. This routing mechanism can be applied to express preferences for desirable nodes and paths, or to restrict traffic to trusted nodes enabling trust and security aware routing. A proof-of-concept implementation of key concepts, developed as an extension of the BMX6 routing protocol, confirms its feasibility and scalability.

In future work we plan to develop a complete implementation of the routing mechanism over BMX6, including syntax, authentication, signature and engine for policy specifications, definition of the bootstrapping procedure. In particular we are interested in further exploring the characteristics and limits of the protocol over larger and more realistic scenarios with the support of the Community-Lab testbed for a detailed evaluation.

3.4 A Monitoring System for Community-Lab

The next research topic corresponds to a research on a distributed monitoring and self-management system designed with the requirements of a distributed testbed such as Community-Lab in mind[35]. It introduces self-management algorithms that try to detect and react to anomalies and correct problems based on locally applied policies. The section describes the resulting architecture, the lessons from initial experiments, and a demo is going to be presented at the conference.

The challenging environment of community networks needs a careful evaluation of experimental data to understand application behaviour and spot any misbehaviour or anomalies. This work focuses on demonstrating a monitoring system tailored to meet the specific requirements of testbed and proposes an architecture for self management to automate management. This demonstration aims to present the current status of the monitoring system, the data gathered and also invite others to experiment with the data generated by the monitoring system.

The experience (and demo) of the monitoring system shows the various metrics that are monitored and how it helps to identify problems in Community-Lab. The associated demonstration aims to present the current status of the monitoring system. In a remote access to the Community-Lab testbed the demo shows the various metrics of the research devices that are monitored and also visualises them. It gives the audience a clear idea on how researchers can monitor the node and experiments along with synthesized metrics to understand the characteristics of the experiment and identify problems if any.

For more in-depth information, we refer to the paper which will be publicly available after the conference.

3.5 NHDP Dual Stack Support

RFC 3626 and OLSRv2 define both an IPv4 and IPv6 mode of operation, but neither protocol specify the details how to support both IP protocol versions at the same time. While the older RFC3626 protocol explicitly uses the IP version of the incoming UDP packets to detect the IP version of the transported messages, the new RFC5444 packet format of OLSRv2 allows to explicitly state the address length of each message within the packet.

While the new packet format allows the option of a dual stack capable OLSRv2 routing agent, it does not address the efficiency or backward compatibility issue.

The easiest backward compatible option would be a single routing agent, sending IPv4 information through IPv4 UDP packets and IPv6 information through IPv6 UDP packets. While this solution is fully backward compatible with non-dual stack implementations, it allows no optimization to reduce the overhead of running both IP versions at the same time.

We have developed a dual stack implementation [18] that is both backward compatible and allows for a more efficient message transport by flooding both IPv4 and IPv6 messages within the same UDP packet.

The OLSRv2 implementation runs a normal RFC6130 (NeighborHood Discovery Protocol - NHDP) implementation on both IPv4 and IPv6. In addition to the standard information the IPv6 Hello messages contain a TLV that includes the IPv4 originator address of the node. This information allows dual stack-capable implementations to detect each other among their direct neighbours.

By keeping a list of implementations for each interface (ipv4 only, ipv6 only, split ipv4 and ipv6, dual stack), dual stack implementations can now optimize the flooding of OLSRv2 control traffic. If

an interface has no neighbour that needs IPv4 messages in IPv4 UDP packets, both IP types can be forwarded within the same IPv6 UDP packet (and vice versa for “no IPv6 dependency”). This allows a mesh network of dual stack capable nodes to only use a single UDP packet to flood both IPv4 and IPv6 message while falling back to splitting messages among IPv4 and IPv6 UDP packet if necessary.

We plan to write an IETF draft about this extension of OLSRv2 and will publish this draft on the IETF Manet list.

4 Conclusions

This deliverable has presented the progress in Work Package 4 on task T4.1: Experimental research on testbed for community networks in year 2.

The first chapter describes studies on the challenges which stem from the scale, heterogeneity and limited resources in the infrastructure the CONFINE community-lab testbed is built on. A study of a single community network is presented, together with a survey of community networks worldwide, followed by improvements to the OLSRv2 protocol and solutions developed inside a community network.

The second chapter shows how link layer and ETT cross-layer metrics can help overcome a number of challenges from the first chapter. A cross-layer analysis is proposed, to take the different layers and their interaction into account.

The third and final chapter proposes general and routing protocol specific self-management techniques as a different approach to tackle the challenges in community networks. A monitoring system is introduced, as well as improvements to the OLSRv2 neighbour discovery protocol.

Overall, the research reported in this deliverable presents good progress and shows a number of interesting opportunities to further improve the testbed and its community network infrastructure.

Bibliography

[1] Davide Vega, Roc Meseguer, Leandro Navarro, and Felix Freitag, "Measurement and analysis of a large-scale wireless community network," Unpublished report. 1.1.1

[2] Christoph Barz and Henning Rogge, "Improved community network node design using a dlep based radio-to-router interface," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*. IEEE, 2012, pp. 636–642. 1.1.1

[3] Llorenc Cerda-Alabern, Axel Neumann, and Pau Escrich, "Experimental evaluation of a wireless community mesh network," in *The 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'13*, 2013. 1.1.2

[4] Emmanouil Dimogerontakis, Ivan Vilata, and Leandro Navarro, "Software defined networking for community network testbeds," in *Accepted for the Second International Workshop on Community Networks and Bottom-up- Broadband (CNBuB)*, 2013. 1.1.3

[5] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008. 1.1.3

[6] "Poxy," https://github.com/emmdim/Poxy. 1.1.3

[7] "Pongo," https://github.com/emmdim/Pongo. 1.1.3

[8] Avonts Jeroen, Braem Bart, and Blondia Chris, "A questionnaire based examination of community networks," in *Accepted for the Second International Workshop on Community Networks and Bottom-up- Broadband (CNBuB)*, 2013. 1.2

[9] "Ubiquiti networks," http://www.ubnt.com/. 1.2

[10] "MikroTik routers and wireless," http://www.mikrotik.com/. 1.2

[11] "Qualcomm Atheros," http://www.atheros.com/. 1.2

[12] "The OpenWrt linux distro," https://openwrt.org/. 1.2

[13] "RouterOS, the operating system of RouterBOARD," http://www.mikrotik.com/software.html. 1.2

[14] "AirOS, the operating system of Ubiquiti," http://www.ubnt.com/airos. 1.2

[15] T. Clausen and P. Jacquet, "Optimized Link State Routing protocol (olsr)," 2003. 1.2

[16] Y. Rekhter, T. Li, and Hares S., "A Border Gateway Protocol 4," 2006. 1.2

[17] "The Better Approach To Mobile Adhoc Networking," http://www.open-mesh.org/projects/open-mesh/wiki. 1.2

[18] C. Barz, J. Niewiejska, and H. Rogge, "Nhdp and olsrv2 for community networks," in *Proceedings of the 2013 IEEE 9th International Conference Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2013. 1.4, 3.5

[19] "Wind," http://wind.awmn.net/. 1.6.2.1

[20] "Wind - wireless nodes database," http://wind.cube.gr/. 1.6.2.1

[21] "Nagios," http://www.nagios.org. 1.6.4

[22] J. D. Case, M. Fedor, M. L. Schoffstall, and J. Davin, "Simple network management protocol (SNMP)," 1990. 1.6.4

[23] Richard Draves, Jitendra Padhye, and Brian Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 114–128. 2.2

[24] H. Rogge and E. Baccelli, "Packet sequence number based directional ett metric for olsrv2 (draft-rogge-baccelli-olsrv2-ett-metric-02)," 2013. 2.2

[25] Llorenç Cerdà-alabern, "On the Topology Characterization of Guifi.net," in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2012)*, 2012. 2.3

[26] Christos Gkantsidis, Milena Mihail, and Ellen Zegura, "Spectral analysis of Internet topologies," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications.*, vol. 00, no. C, 2003. 2.3

[27] M Faloutsos, P Faloutsos, and C Faloutsos, "On Power-Law Relationships of the Internet Topology," *ACM SIGCOMM Computer Communication Review*, 1999. 2.3

[28] Y Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, Mar. 1996. 2.3.1.3

[29] Edoardo Amaldi, Antonio Capone, Matteo Cesana, Ilario Filippini, and Federico Malucelli, "Optimization models and methods for planning wireless mesh networks," *Computer Networks*, vol. 52, no. 11, pp. 2159–2171, 2008. 2.3.2.3

[30] Suvrajeet Sen, Robert D Doverspike, and Steve Cosares, "Network planning with random demand," *Telecommunication Systems*, vol. 3, no. 1, pp. 11–30, 1994. 2.3.2.3

[31] "Ninux community network," http://ninux.org. 2.3.3

[32] Xavier León and Leandro Navarro, "Incentives for dynamic and energy-aware capacity allocation for multi-tenant clusters," in *Economics of Grids, Clouds, Systems, and Services*, pp. 106–121. Springer, 2013. 3.1

[33] Davide Vega, Roc Meseguer, Felix Freitag, and Sergio F Ochoa, "Effort-based incentives for resource sharing in collaborative volunteer applications," in *Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on*. IEEE, 2013, pp. 37–42. 3.2

[34] Axel Neumann, Leandro Navarro, Roger Baig, and Pau Escrich, "Receiver-driven routing for community mesh networks," . 3.3

[35] Navaneeth Rameshan, Leandro Navarro, and Ioanna Tsalouchidou, "A monitoring system for community-lab," November 2013. 3.4