UNIVERSITÀ DEGLI STUDI DI CATANIA

DIPARTIMENTO DI INGEGNERIA
ELETTRICA, ELETTRONICA E INFORMATICA

DOTTORATO DI RICERCA IN INGEGNERIA
INFORMATICA E DELLE TELECOMUNICAZIONI

XXIV CICLO

On Location-Awareness in P2P Wireless Mesh Community Networks

CORRADO RAMETTA

IL COORDINATORE
Prof. O. Mirabella

IL TUTOR
Prof. S. Palazzo

Contents

List of Figures

Abstract

The success of experiences such as Seattle and Houston Wireless has attracted the attention on the so called wireless mesh community networks (WMCNs). These are self managing wireless networks in which each node acts as a router able to forward data flows towards the designated destination directly or via multi hop paths. WMNs, whose success is due to their capability in building cost effective and highly scalable wireless networks, are spontaneously deployed by users willing to share communication resources as well as multimedia contents. Become popular since the introduction of cheap wireless technologies such as IEEE 802.11, WMCNs represent a promising framework aiming at reducing the digital divide between town and countryside and, consequently, promoting social communication and business advancements in rural areas.

Peer-to-Peer networks, due to their capacity of providing a good substrate for large scale content-resources sharing and distribution applications, represent an interesting and promiscuous research area of the ICT and could play a key role in the diffusion

of mesh paradigm meeting the growing need of communication and resource sharing among people anywhere and anytime.

Extending the P2P paradigm to wireless networks presents several difficulties due to their dynamic, multi hop and often power and computational constrained nature. While ad hoc and sensor networks represent very challenging environment due to, respectively, the high mobility and the power-computational constrains, for wireless mesh there are not so many restrictions. Nevertheless, some considerations must be taken into consideration. We provide some fundamental guidelines in designing P2P overlay schemes for WMCNs reaching the conclusion that a hierarchical, structured, and distributed-hash-table (DHT) based architecture, exploiting location-awareness, could represent a suitable solution able to guarantee high resilience, fault tolerance, flexibility and scalability, at the cost of supporting a moderate overhead for overlay and nodes' churning management.

According to this idea, we studied and evaluated a location-aware DHT-based P2P scheme, called Georoy. We also introduced some improvements allowing its applicability to wireless mesh networks and opportunistic rural scenarios.

Keywords: wireless mesh community networks, opportunistic networking, peer-to-peer overlay, information retrieval, location-awareness, distributed hash table, lookup performance.

1 Background

In this chapter we will introduce some fundamental concepts representing the know-how of the dissertation.

1.1 Wireless Mesh Networks

Before discussing about the wireless mesh networks paradigm, we need introduce the concept of mobile ad hoc network (MANET). It consists of wireless nodes connected together over a wireless medium and able to freely and dynamically self-organize into arbitrary topologies allowing users and devices to communicate without any preexisting communication infrastructure. This concept is not new but it is remained circumscribed to a few of niche sectors such as tactical networks employed essentially for military or specialized civilian applications (disaster recovery, rescue missions, etc.). The requirements for this kind of applications are far from the real users' requirements. Indeed, military and civilian specialized applications require lack of infrastructure, instant and self-organizing deployment, high fault tolerance, high resiliency, QoS,

security and so on. They are tailored for specialized uses and their cost rather is a crucial issue. On the other hand, the above cited approach is far from the general porpoise scenario the ordinary user is interested to, the latter consisting of a limited number of users/devices willing to share some information, computational/storage resources, or access to the Internet. In this case the lack of infrastructure is not a must and the focus in more specifically on a more pragmatic "opportunistic ad hoc networking" [8] in which multi hop ad hoc networks are not isolate self-configuring networks, but represent a flexible and low cost extension of wired infrastructure networks able to guarantee interconnection among the wireless devices as well as the Internet access. This networking paradigm is the conceptual base of the wireless MESH networks (WMNs).

1.1.1 Network architecture

As shown in Figure 1-1, WMNs are built on a mix of fixed and mobile nodes interconnected via wireless links to form a multi-hop wireless networks usually providing access to the Internet by means of gateways physically connected to the wired Internet backbone. WMNs consist of two types of nodes: mesh routers and mesh clients. Mesh routers have minimal mobility and do not suffer from power constrains; they are equipped with more than one wireless interfaces in order to provide MESH access to the clients

and forwarding data packets from source to destination or towards the Internet in a multi hop manner. The integration with other networks such as Internet, cellular, IEEE 802.16, sensor networks, etc., can be achieved through the gateway and bridging functions implemented in the mesh routers. Mesh clients are mobile or stationary devices that that can form a client mesh network among themselves and with mesh routers. Mesh clients are usually more simple devices built based on general porpoise computer, notebook, smartphone systems instead, mesh routers, are built based on dedicated computer systems.

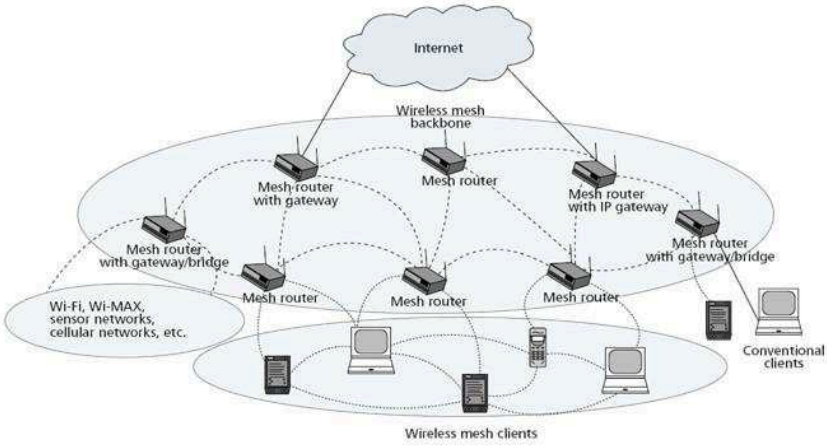

Figure 1-1: Wireless Mesh Network architecture [7].

Notwithstanding this routers and clients are usually built based on similar hardware platforms: clients have necessary functions for mesh networking and, thus, can also work as a router. However they do not implement gateway and bridge functions.

Akyildiz et al. in [7] classify the architecture of WMNs into three main groups:

1. **Infrastructure/Backbone**: consists of mesh routers forming an infrastructure, well known as backbone, for clients that connect to them. The routers form a mesh of self-configuring and self-organizing wireless links among themselves. Routers also implement gateway/bridge functionalities thanks to which it is possible connecting the mesh net to other kinds of network such as cellular, WiMAX, and above all the wired Internet backbone. This kind of architecture is the most successful one and it represents the technological base for deploying wireless rural community and neighborhood networks. Internet access, contents and resources sharing among participants are the killer applications of this network paradigm.
2. **Client WMNs**: client meshing provides peer-to-peer connectivity among mesh clients. In this type of architecture clients build the network by means of self-organizing links among themselves. Client devices must be able to route data packets towards the designated destination also exploiting multi hop path between source and destination of an end-to-end communication. Thus, client devices must implement additional functions such as routing and self-organization.

3. **Hybrid WMNs**: this architecture comprises both the two above mentioned paradigm. Mesh clients can communicate each other by means of the client mesh they build; furthermore they are also able to gain connectivity towards other networks such as Internet, WiMAX, cellular, etc, thanks to the infrastructure network made up of mesh routers including gateway/bridge functionalities. This approach is the same shown in Figure 1-1.

1.1.2 Mesh benefits and criticalities.

WMNs success is due to their capability in building cost effective and highly scalable wireless networks, representing a solution for the easy and fast deployment of connectivity architecture and ubiquitous Internet access. Summarizing, we can highlight the following peculiarity that characterize wireless mesh:

Cheap and fast installation procedures. Currently, the most popular solution to offer Internet connectivity in outdoor environment consists in deploying the so called hot spot. The latter are essentially wireless access point exploiting the IEEE 802.11 standard and are designed to provide a limited number of user devices with connectivity, moreover at very short distances. Such architecture can be extended only introducing a sufficient number of access points directly connected to the wired backbone; in the

light of this, it appears clear that this kind of architecture is not fast and cheap deployable. On the other side, mesh architecture guarantee connectivity to a great number of devices and is able to gain Internet access using only a limited number of mesh routers directly connected to the wired backbone. In such a way it represents a cost effective and fast-deployable solution to offer communication possibilities in a vast range of situations and scenarios.

Large scale deployment. Because of the possibility to exploit multi hop path from a source to a destination node, mesh networks can cover vast outdoor environment; furthermore, taking advantage of fixed powered wireless routers they can implement sophisticated transmission techniques improving the transmission data rate and the wireless links covering distance respect with the conventional, usually power constrained, WLAN technologies involved in wireless ad hoc networks.

Reliability. WMNs guarantee multi hop connectivity between end users using hop-by-hop forwarding towards the destination. Using this approach multiple paths can be detected between two nodes involved in a communication so WMNs exhibit high resilience and fault tolerance against node and link failures.

Self-configuring and management. Exploiting a peer-to-peer paradigm to build the wireless distributed system means taking advantage of all the features of this kind of architecture, essentially self-configuration, self-management and, consequently, the capability to effectively react to system changes (e.g., nodes joining or leaving the network) and failures (due to power constrains, radio interference, obstacles between antennas, etc.). These features makes WMNs fast and simply deployable.

What above cited is not an exhaustive list of benefits characterizing the wireless mesh technology but want to provide some key factors that permit to justify because this architecture represents a promising research and application field in the information and communication technology area.

Despite all the benefits illustrated, WMNs are a challenging environment because of several aspects that it need take into consideration during the network design:

- **Obstacle to scalability**. wireless mesh are based on multi-hop communication among nodes in order to connect source and destination. There are several research papers asserting that multi-hop environments suffers from scalability issues, i.e., when the size of network increases the network performances decreases significantly; transport protocols may loose connection, routing protocols are not able to establish paths between source

and destination node, and MAC protocols may experience significant throughput degradation, unfairness and starvation. Centralized medium multiple access control protocols such as TDMA or CDMA, due to their general requirements such as time synchronization or code management, are difficult to implement in a decentralized structure.

- **Criticality of QoS management**. WMNs are distributed communication architectures thus guarantying QoS parameters represents a very challenging issue considering that, in addition to end-to-end transmission delay and fairness, more performance metrics such as delay jitter, aggregate and per-node throughput, packet loss rate, etc., must be taken into consideration by communication protocols.
- **Security**. Security represents a key factor for the adoption of a network technology rather than another. In particular, WMNs suffer from security problems commons to all the other wireless and distributed technologies. In a wireless environment each node can listen the communication of each other if it is in its coverage area. so, it is indispensable introducing efficient schemes able to guarantee security and privacy in data transmissions. Although many algorithms have been proposed for wireless LANs, they are still not ready for WMNs due to

the lack of a centralized trusted authority able to distribute a public key in a WMN. On the other side, the existing security schemes proposed for ad hoc networks could be adopted for WMNs, but they are still not mature enough to be practically implemented. Nevertheless, several solutions have been introduced to improve this critical aspect such as secure and anonymous routing (e.g. ANODR and Secure Routing Protocol), secure MAC, etc.

- **Capacity of WMNs**. The study of the capacity of a WMN represents a very hard challenge to be solved. There are too many factors influencing it such as network topology, traffic pattern, node density, number of radio and or channel employed, transmission power level, node mobility, physical channel modeling, radio interferences, and so on. Under this perspective it appears clear that understanding all the relationship between network capacity and the above factors is a task that unlikely will be concluded exhaustively.

1.1.3 Application Scenarios

Research and development efforts in wireless mesh technology are due to the several applications for which it represents a promising and effective networking solution. In the following we will present some of these application scenarios.

Public Internet access and community networking. Nowadays the most popular architecture for network access is based on cable or DSL connected to the Internet. Only the last-hop is performed using wireless technologies, and in particular the IEEE 802.11 standard. Exploiting this approach, where each home have to use an individual connection towards the Internet, presents several drawbacks:

- Even if the information concerns the community it has to be sent and received through Internet;
- The access to the Internet is expensive and not fast deployable;
- Vast areas of the community are not covered by wireless services;
- Each home has only one available path to access the public network, so the architecture suffers from single points of failure issue;
- Many interesting applications such as computational-storage-data resources sharing among the community members are not available.

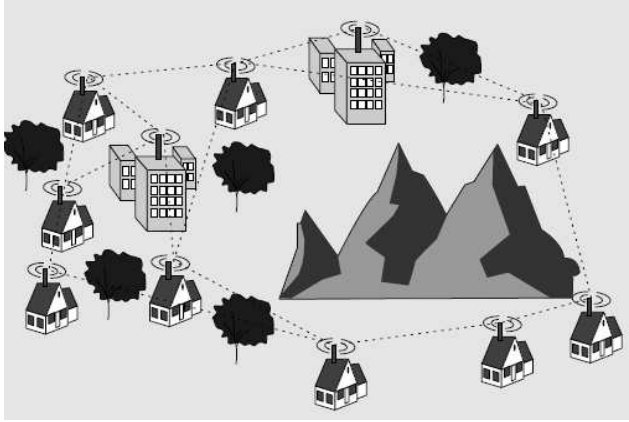

Figure 1-2: Broadband Internet access and community networking [8].

WMNs enhance this kind of architecture providing users with a cheap, fast deployable, resilient and fault tolerant network where they can both access the Internet and share resources in a P2P manner. On the base of the above considerations it appears clear that WMNs represent the ideal solution to solve the digital divide issue between town and countryside providing an excellent framework for delivering broadband services to such areas. Indeed, the necessity of connecting to the wired backhaul network only a little part of the mesh routers permits a cost effective deployment even with a limited number of subscribers, as found in rural or scarcely populated urban areas.

Spontaneous networking and P2P communications. There are several circumstances requiring a fast deployable, effective and fault tolerant communication architecture. This aspect is

particularly relevant if we think to the public safety: the 9/11 events and the consequently war against terrorism as well as the recent natural disasters have created an urgent demand for wireless network connectivity allowing fast, simple and high bandwidth communications for military and police forces, rescue services, fire brigades, medical corps, etc. Solutions based on cellular technologies, even though they guarantee near ubiquitous coverage and high-mobility speeds, present several drawbacks: the network infrastructure is not always available, (e.g., in facing up to natural disasters the cellular infrastructure can be damaged and only partially available); deploying an ad hoc cellular architecture is very expensive and it does not represent a viable solution in tactical and military operations; guaranteed data rate is often limited, even lower than a dialup connection, so it is not suitable for image, voice and video transmissions. Vice versa, mesh networks successfully address the spontaneous (emergency, disaster recovery, tactical) networking issues offering an effective, scalable, easily deployable solution.

Among the spontaneous networks we can also mention P2P communications. For a group of people holding devices with wireless networking capabilities, e.g. laptops, PDAs, tablets, smartphones, etc, P2P communications anytime and anywhere represent a must. For their characteristics WMNs are able to meet this demand in a very simple and cheap manner.

Enterprise networking. Under this paradigm we can consider small (office case) and medium (the case of offices in a building) size as well as large scale (the case of offices in multiple buildings) networks. Currently, standard IEEE 802.11 wireless networks are widely used in various offices but, if this architecture could result effective for small size office or enterprise, it results inadequate for medium to large scale enterprises or offices. Standard 802.11 access points need to be directly connected to the wired LAN for providing Internet access and guarantying end-to-end communication among users of the same enterprise belonging to different offices each far from the other. If the access points are replaced by mesh routers Ethernet wires can be eliminated and multiple backhaul access modems can be shared by all terminals of the network, improving resilience, fault tolerance and resource utilization. Furthermore, mesh nets can adapt easily as the size of enterprise expands. Obviously, the service model above illustrated can be applied to many other public and commercial service networking scenarios such as airports, hotels, hospitals and medical centers, shopping centers, and so on.

Transportation systems. WMNs are a viable solution for meeting two important demands related to transportation systems:

- Realizing intelligent transportation systems, i.e. integrated public transportation systems built to be safe, efficient and secure. Wireless mesh represent a flexible solution to

realize the information delivery system necessary to control and manage transportation services. WMNs allow driver communications, remote monitoring of in-vehicle security video, information about traffic congestions, pollution control, service awareness for passengers.

- Providing passengers with Internet connectivity. Often Internet access is limited to stations and stops thanks to 802.11 technology, but it should be useful introduce this capability also during the travel. Also in this case mesh technology offers a suitable solution able to guarantee this kind of service. Obviously, a mesh architecture for this porpoise needs two key techniques: a high-speed mobile backhaul from a vehicle (car, bus, train, etc.) to the Internet and mobile mesh networks within the vehicle.

1.1.4 Standards on Wireless Mesh Networks.

Actually, wireless technology is most popular in one hop network architectures. Typically, it is used to provide Internet access to fixed or mobile devices equipped with compliant wireless cards and connected to a wireless access point. The IEEE 802.11 standards and the WiFi Alliance played a key role in the success of this kind of architecture but a lot of efforts must be addressed in the field of mesh networking. 802.11 a/b/g/n can be used in ad hoc manner but it presents several drawbacks in multi hop connectivity,

as explained above, thus we can consider it far from to be a suitable solution for mesh networking. The lack of a well established standard and the growing interest in wireless mesh applications has boosted industrial efforts to develop proprietary solutions to make WMNs a reality. Some vendors initially focused on products based on IEEE 802.11 technologies coupled with proprietary software solutions making these systems incompatible one with other. Several other vendors adopted solutions based on proprietary radio technologies considering the 802.11 standards not adapted for this kind of approach.

In despite of this, open standards are essential for industry and for a wide deployment of a technology because they enable economies of scale, which bring down the cost of equipment and ensure interoperability. For these reasons several IEEE standard groups are actively working to define specifications for wireless mesh networking. Because of the complexity of the problem different task groups have been created to define the requirements and provide the related solutions for the different mesh networking architectures. In the following sections we will present the results of these efforts in the different areas of interest: wireless personal area network (WPAN), wireless local area network (WLAN) and wireless metropolitan area networks (WMAN).

1.1.4.1 IEEE 802.15.5

Wireless personal area networks (WPANs) are a very interesting solution for home, office and wireless sensor networks. WPANs are characterized by short distance among nodes and low power consumption. On these requirements focuses the activity of standard groups such as IEEE 802.15, Bluetooth Special Interest Group (SIG), WiMedia Forum, UWB Forum and so on.

In particular, we will focus on the IEEE 802.15 standard family that contains many task group covering almost all scenarios where WPANs are involved. It follows a brief description of the different standards:

- **802.15.1**: studies PHY and MAC specifications for wireless connectivity of fixed, portable and mobile devices belonging to the so called personal operating space (POS), that is, a space enveloping the person and extending up to 10 m in all directions. It is based on Bluetooth technology and, according to this, it works in the 2.4 GHz ISM frequency.
- **802.15.2**: because the previous standard operates in the same 2.4 GHz band of 802.11 products, the work of this new task group focused on specifying the coexistence mechanisms between wireless PAN and LAN and other networks working in the same unlicensed frequency range.

- **802.15.3**: this task group was created to specify a new MAC and PHY for WPANs able to support high rate applications such as digital imaging and multimedia achieving scalable data rate from 11 Mbps to 55 Mbps. This working group originated two different amendments, “a” and “b”: the former focused on correcting and revising the basic 802.15.3 standard, the latter focused on increasing the transmission data rate developing ultra wide band (UWB) based wireless PAN in order to support high rate multimedia traffic. The activity of group “a” was withdrawn but the researches on this field has never stopped thanks to the efforts of two industrial consortiums called WiMedia Forum and UWB Forum.
- **802.15.4**: this standard defines PHY and MAC layers for low rate wireless PANs. Regarding the former, two types are specified: 868/915 MHz direct sequence spread spectrum (DSSS) and 2450 MHz DSSS, the first achieving a data rate equal to 20/40 Kbps and the second a data rate up to 250 Kbps. MAC layer adopts a CSMA/CA approach, in both star and peer-to-peer topologies.

In all the above mentioned standards mesh topologies are not considered. Extending WPANs with mesh capabilities is the focus of **802.15.5** task group. The latter aims to provide a recommended practice rather than a mandatory standard for defining an

architectural framework allowing WPAN devices to be interoperable in a stable and scalable wireless mesh topology.

In 802.15.5 we can distinguish three types of PAN device: PAN coordinator, coordinator, and end device. End devices are connected to the related coordinator forming a star topology as that employed in other 802.15 PANs while coordinators are connected to each other through a mesh topology. Defining the latter means, above all, redesigning the MAC sublayer and this is the major effort of 802.15.5 standard group. While other 802.15 standards only focus on PHY and MAC without any remark on the routing procedures, specifying a routing protocol is one of the most important tasks for the 802.15.5 task group. In particular, the MAC has been enhanced based on that of other WPANs and new routing function has been added on top of the enhanced MAC protocol. The introduction of routing capabilities is a key feature to guarantee the interoperability of products belonging to different vendors. Since, as mentioned above, WPANs address both low rate and high rate wireless networks having different PHY and MAC sublayers, the 802.15.5 task group is currently working on separate specifications for them. However, the protocol stack of these two types of network is the same and the proposed routing capabilities are based on the meshed tree approach.

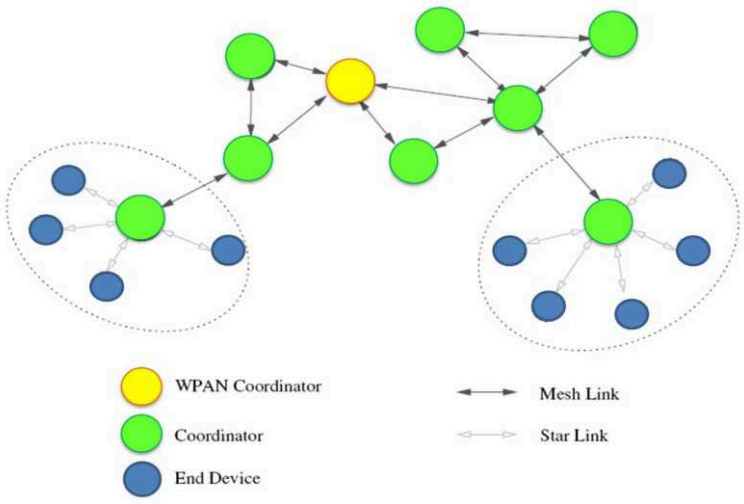

Figure 1-3: 802.15.5 meshed wireless PANs (adapted from [10]).

The tree defined in the proposal is called adaptive robust tree (ART) and is shown in Figure 1-4. Three phases are defined in ART:

- **Initialization**: during this phase, nodes joining the network build the ART tree. Logical addresses are adaptively assigned during the tree formation procedure to reflect the actual network topology; furthermore each node keeps an ART table (ARTT) to track its branches and, to each branches is assigned one or more blocks of consecutive addresses. In particular, the ART tree formation is divided into two steps: association and address assigning. During the first one, beginning from the root, nodes gradually join the network forming the tree.

But this is not sufficient to make the tree an ART. To address this, it needs that each node has a logical identifier. After the tree reaches its bottom, a down-top procedure begins in order to calculate the number of nodes for each branch, as illustrated in Figure 1-4. When the procedure ends, each node of the tree can indicate a desirable number of addresses.

- **Operation**: new nodes are still allowed to join the network and, for each substantial change of either the number of nodes or the network topology, the tree need to be reconfigured.
- **Recovery**: when the tree is broken then the recovery procedure is triggered, involving only the affected part of the tree. The ART is constructed in such a way that tree repair and recovery can be accomplished without changing any assigned identifier.

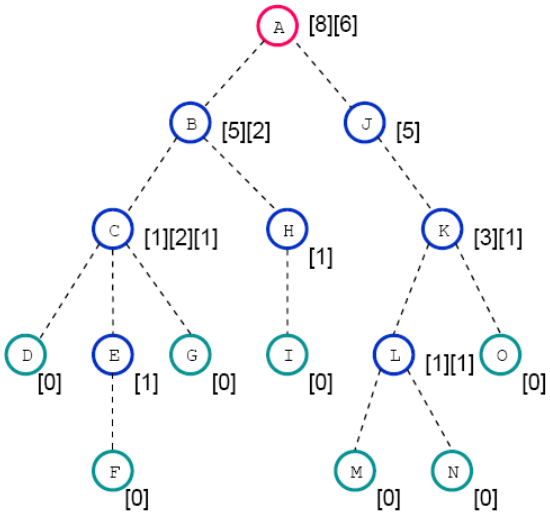

Figure 1-4: Calculation of number of nodes along each branch [66].

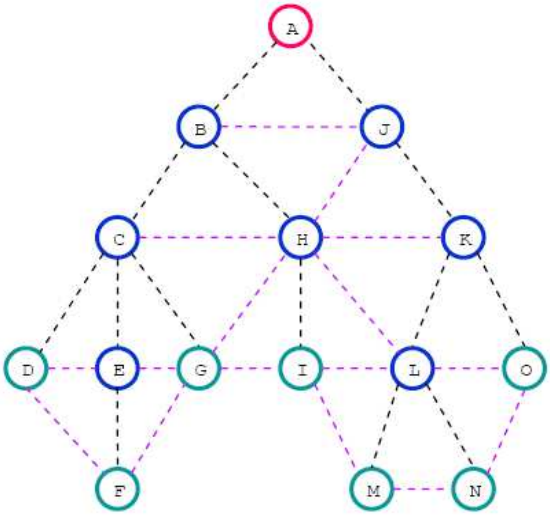

Figure 1-5: Meshed ART [66].

By exploiting this procedure, it is possible to build a meshed ART (MART) on top of an ART. This is realized by using additional links (shown in Figure 1-5 as magenta lines) in such a way that the network looks more like a mesh than a tree even if from each node's point of view the network is still a tree. By forming a MART it is possible to both route packets through shorter paths and remove single points of failure.

1.1.4.2 IEEE 802.11s

The IEEE 802.11 [67] family is currently the most popular and successful wireless networking standard for wireless LANs. It define the physical layer and MAC sublayer for the devices used in WLAN networking. IEEE 802.11 a/b/g/n are based on a structure consisting of a device called Access Point (AP) to which end user devices or stations (STAs) connect for accessing network services. The set of STAs managed by the related AP is called basic service set (BSS). Furthermore, the above mentioned standards also permit another kind of networking: the independent basic service set (IBSS), well known as ad hoc network, where each device can communicate with others without APs. A set of BSS, interconnected by a distribution system (DS), usually realized by wired technology, forms an extended service set (ESS). Under this perspective, we cannot define this architecture because the distribution system is generally a wired LAN. The IEEE 802.11

family continue to advance with the introduction of various amendments but they are still limited because of their dependency upon the wired technology to constitute the distribution system. Using 802.11 as mesh technology can be a solution (e.g., we can consider a mesh network realized using 802.11 b/g/n as one hop access networks coupled with 802.11 a in ad hoc mode for realizing the multi hop mesh backbone) but 802.11 standards primarily aim at fulfilling one-hop communication needs and, as a consequence of this design choice, they are affected by the problems of throughput degradation, unfairness and starvation when applied in multi hop architectures. With the aim of solve this problem and obtain an effective solution for wireless distribution systems (WDS), a separate TG called IEEE 802.11s was formed in May 2004.

The activity of 802.11s TG focuses on the specification of a new protocol suite for the installation, configuration and management of WLAN mesh. Its implementation is based on the PHY layer of 802.11 a/b/g/n operating in the unlicensed 2.4-5 GHz frequency bands, introducing the major novelties in the MAC sublayer design.

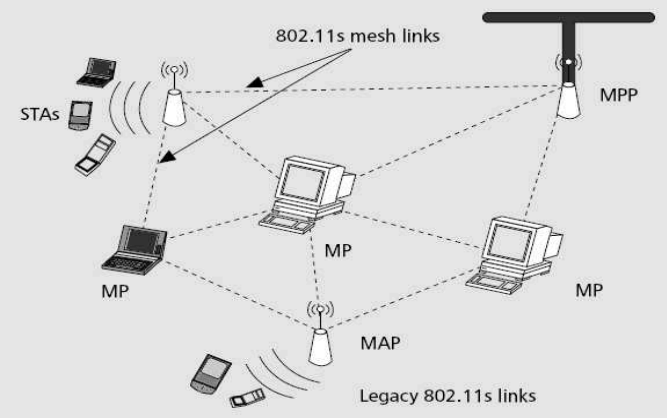

Figure 1-6: IEEE 802.11s network architecture [9].

In particular, MAC 802.11s will introduce a path selection protocol for routing data in the mesh topology instead of the routing protocol. The proposed architecture is depicted in Figure 1-7 regarding which we can provide the following definitions:

- **Mesh Point (MP)**: device that support mesh-relay functions including neighbor discovery, channel selection, association with neighbors;
- **Wireless Distribution System (WDS)**: is the set of MPs and the related wireless links they form;
- **Mesh Access Point (MAP)**: is a specific mesh node that include the capabilities of an access point thanks to which other 802.11 compliant devices can be connected to the mesh network. Usually it implements 802.11s as mesh standard to join the mesh and 802.11 b/g/n to play the role

of access point providing connectivity to other legacy user terminals:

- **Mesh Portal Point (MPP)**: is the more complex device including all the capabilities of a mesh point but providing connectivity towards other meshes or access to the Internet: it acts as a gateway towards the external network technologies.

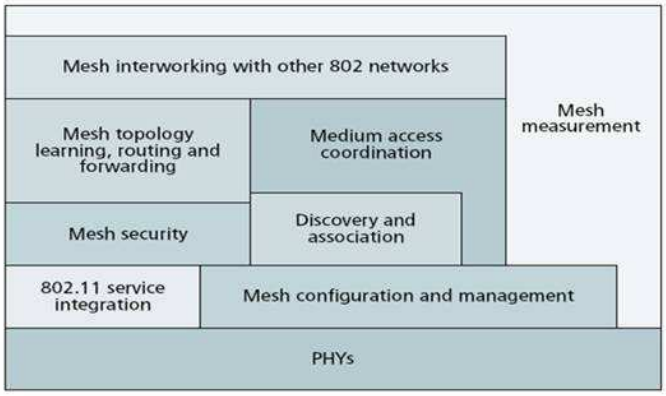

Figure 1-7: Architecture of MAC 802.11s [9].

The most important novelties involving the MAC design are illustrated in Figure 1-7 and they include:

- Topology learning;
- Routing and forwarding;
- Medium access coordination in the distributed architecture;

- Mesh configuration and management;
- Mesh measurements;
- Security functions.

1.1.4.3 IEEE 802.16 mesh mode

The IEEE 802.16 [68] working group defines the physical layer and the MAC sublayer standards for wireless metropolitan area networks (WMANs). This standard, associated with Worldwide Interoperability for Microwave Access (WiMAX), was designed to operate in the licensed 10-66 GHz frequency range requiring line-of-sight (LOS) towers, called base stations (BSs), covering up to 5 Km, using an architecture similar to that employed in the cellular networks. It was defined to meet the need for a backhaul network for the broadband wireless access at much lower cost compared with the wired counterparts, such as DSL and cable. The standard was initially created for point-to-multipoint architecture aimed to provide higher data rate (up to 75 Mb/s) for each subscriber station (SS). Basic standard was expanded creating several task groups, from “a” to “g”, to address:

- **a**: addition of mesh capabilities;
- **b**: providing quality of service (QoS) feature;
- **c**: supporting interoperability;
- **d**: extension of physical layer;

- **e**: supporting mobility;
- **f**: supporting multi hop capabilities in standard e;
- **g**: providing efficient handover and QoS.

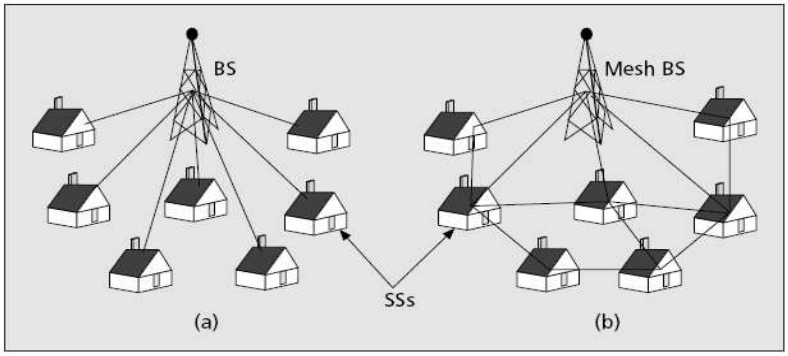

Figure 1-8: IEEE 802.16 in (a) PMP mode and (b) mesh mode [9].

The IEEE 802.16a standard adds the mesh mode to the PMP defined in IEEE 802.16.the new standard operates in the licensed and unlicensed frequencies of 2-11 GHz that allows non-line-of-sight (NLOS) communications, obtaining a coverage area of 50 Km. While in the PMP each SS must be connected to a BS, in mesh mode a SS can communicate directly with each other without the need of a BS. Thus, a SS in mesh mode serves as a router able to forwarding the data traffic originated by a neighbor towards the target BS that connects the mesh to other external networks.

Active nodes belonging to the mesh network send periodically MSH-NCGF (mesh network configuration) messages exploited by new nodes wanting to join the mesh for synchronizing with the

existing network. A new node actively scans to hear the MSH-NCFG messages and, once received one or more, it establishes synchronization and initiates the entry procedure sending a MSH-NENT (mesh network entry) message. After authorization, it receives a 16-bit node identifier from the mesh BS.

Medium access is based on two types of TDMA scheduling mechanisms:

- centralized scheduling: the BS manages the communication resources for all SSs within a certain hop range;
- distributed scheduling: all nodes coordinate with each other for accessing the channel, including the mesh BS.

1.2 Opportunistic Networking

An opportunistic network [12] is a type of challenged network where intermittent network contacts are met and link performance are variable and unstable. In general in these networks stable end-to-end paths do not exist since nodes can be isolated most of the time and paths may frequently break up. To cope with these problems while supporting communications, a store/carry-forward approach can be used where intermediate nodes keep the message while the connectivity is down. This requires that applications are delay-tolerant [58]. Moreover, the use of an opportunistic paradigm

allows to foresee a process of resource propagation during occasional intercontacts between nodes.

ZebraNet [13] is an example of DTN networking, which tracks animal movements across a wide area. Collars carried by animals work like peer-to-peer devices which communicate to deliver logged data to monitoring centers. Opportunistic networking is also dealt with in an analytical perspective in the Pocket Switched networks [14] where intercontact times among pairs of nodes are analyzed in real human mobility scenarios. Similar studies aimed at characterization of social interactions have been also carried out at MIT in the context of the reality mining project [15]. Also the Haggle project [16] proposed a networking architecture along with a set of protocols and description languages to enable communication in intermittent network connectivity scenarios.

Concerning the network layer, two routing approaches are common in opportunistic scenarios: forwarding and flooding. In forwarding, intermediate nodes relay a single copy of the packet over several hops towards the final destination. The difference among the various forwarding approaches relies in the methodology used for selecting the best path for forwarding data: direct-transmission [17, 18], location-based transmission [19, 20] or using an estimation based approach [21, 22]. The forwarding approach has typically low overhead in terms of packets circulating in the network but can suffer for low packet delivery ratio and long delivery delays. On the contrary, the flooding based schemes are

more robust but can add significant overhead into the network by having multiple copies of a packet traversing the network.

In opportunistic networks, a connection-oriented transport layer protocol such as TCP requires reengineering due to frequent disruptions and intermittent end-to-end connectivity. For example, the Licklider Transmission Protocol (LTP) and its evolutions have been introduced in order to cope with retransmissions in high latency environments such as the challenged ones. Typically, a new protocol layer is required to be identified and located in between the application and transport layers. This protocol, denoted as bundle layer [23, 24], allows each node to act as both a router or a gateway to transfer messages across different regions. In this way the problem of supporting traditional applications where the end-to-end source-destination connections do not exist can be overcome. At the Bundle layer, functionalities of storing/carrying- forwarding are considered and employed for multicast and anycast [25–27].

Finally, concerning the application layer, support for traditional applications such as Web and email is not possible since the underlying transport protocols do not work properly in challenged opportunistic environments. As a consequence, in [28] the use of SMTP proxies is introduced to hide disruptions among users. Emails are thus sent in bundles into the opportunistic network and carried to a mail gateway which forwards and receives the mail between the infrastructured and the opportunistic networks. In [29], an Internet proxy is used to collect search

engines and prefetch web pages. The user query is stored until the mobile node will contact the proxy after a disruption.

1.2.1 Opportunistic networks for developing areas

The success of experiences such as Seattle and Houston Wireless has attracted the attention on the so called wireless mesh community networks (WMCNs). These, become popular since the introduction of cheap wireless technologies such as IEEE 802.11, are spontaneously deployed by users willing to share communication resources (usually to obtain an adequate Internet access) as well as other resources such as data, news, images, music, movies and so on. WMCNs represent a promising framework aiming at reducing the digital divide between town and countryside, which degrades both social communication and business advancements in rural areas.

Opportunistic paradigm becomes critical in challenging scenarios like rural communications in emerging countries like India or Africa, where the lack of an infrastructure makes communications almost impossible. Opportunistic and Delay-tolerant (DTN) communications are thus the natural choice for a networking paradigm where nodes can be disconnected from the Internet for the majority of the time and exchange of data can take very long time.

In emerging countries numerous projects aimed at rural poverty alleviation have been proposed. For example the Sustainable Access in Rural India (SARI) program [56], inaugurated in 2001, consists of disseminating more than 80 rural Internet kiosks distributed in the Madurai area of Tamil Nadu in India. However, not all villages can be served by these kiosks and thus, in parallel, exploiting an opportunistic approach, the Computers on Wheels (COW) project [57] has been carried out in India as well since 2003. In this case, a set of motorcycles equipped with an Internet-connected laptop travel between very remote villages to collect requests for Internet access and support users' communications during the limited time the motorcycle stops at the village.

2 P2P overlay networks

Peer-to-peer networks represent an interesting and promiscuous research area of the so called information and communication technology. This interest is due to their capacity of providing a good substrate for large scale data/content/resources sharing and distribution applications.

There are several works in the field so, the aim of this chapter is providing a classification and a brief description of the most important scheme proposed in the recent years.

For the sake of simplicity we can say that a P2P network is made up of nodes, called peers, that - differently from the client-server paradigm - play symmetric roles in the architecture. They form a self-organizing overlay network overlayed on the network protocol and offering a set of features useful for the applications running over it.

Nodes joining a P2P network can be deployed in a local area or in a wide area. Although they can communicate with each other by means of the network protocol, i.e. the Internet Protocol, this is not sufficient for realizing a P2P architecture. This latter needs the definition of procedures for managing peers and resources, routing the requests for contents, guarantying security, reliability, fault

resiliency and trustworthiness. This is the scope of the overlay network according to which:

- peers joining the network are organized in a graph that can be unstructured or structured, centralized or decentralized;
- peers can look for contents shared by the other peers sending a lookup query message through the network;
- lookup query messages and related replies can be routed towards the destination, i.e. the peer that hold the resource.

In [4] authors propose an abstract P2P overlay network architecture consisting of 5 levels:

- **network communications layer**: describes the network characteristics of the terminals connected. A P2P network can be made up of personal computers connected by Internet, wireless terminals or sensor devices connected in ad-hoc manner;
- **overlay nodes management layer**: this refers to the procedures necessary for building the overlay structure, managing nodes' joining/leaving actions, managing the shared resources, defining and keeping update the routing table for each peer, routing the lookup query requests and replies;

- **features management layer**: deals with the security, reliability, fault resiliency, trustworthiness and content replication aspects;
- **services-specific layer**: represents an intermediate level between application tools and the underlying P2P architecture supporting the application-specific components;
- **application level layer**: consisting of tools, applications and services that are implemented on top of the underlying P2P overlay network.

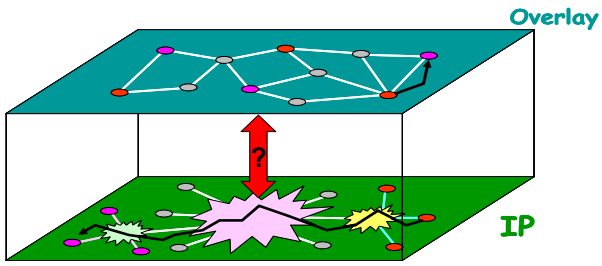

Figure 2-1: the difference between network and overlay layer.

We can classify the overlay network schemes into two great family of algorithms on the base of the organization of the graph: Unstructured and Structured.

Unstructured. peers are organized in a random manner, the graph has not an ordered structure and resources storage and management don't follow any rule. Each peer can manage

whatever content. The random graph, resulting from the building of the network, is generally organized or in a flat or in a hierarchical manner. In flat organizations each node has the same role in building and managing the overlay. In hierarchical case, usually, we can distinguish two types of nodes participating the network: peers and super peers. The former represent nodes that want to share contents or resources, the latter instead play a key role in the building and management of the overlay and provide peers with the procedures necessary to the resource discovery and retrieval.

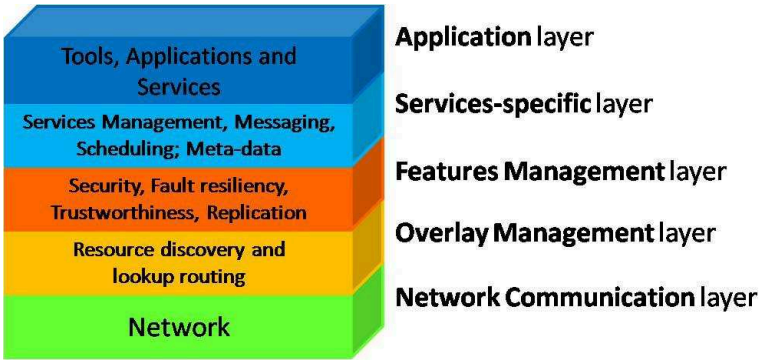

Figure 2-2: Abstract overview of P2P overlay architecture (adapted from [4]).

Unstructured architectures employ flooding or random walks to forward a lookup query (i.e. a packet sent by a peer wanting to discover and retrieval a resource or a content stored in another peer of the network). It is obvious that a such paradigm is effective and suitable only if resources are very popular among participants to the net and, thus, multiple copies of a certain resource are available.

If this is not the case, instead, the lookup procedure necessary to retrieval a content becomes inefficient: lookup query for a rare content has to be sent to a large fraction of peers before reaching the node managing or storing the requested resource. Well known examples of unstructured approaches are: Gnutella [36], Freenet [44], BitTorrent [47], eDonkey2000 [48][49], FastTrack [45], KaZaA [46]. We will discuss them in the following paragraph.

Structured: these architecture are based on the construction of a well defined graph. They also can be classified in flat and hierarchical on the base of the organization of peers participating to the net. Usually, in this kind of approach, peers have a logical ID that permits to identify them within the overlay. IDs are assigned in a logical address space. Shared contents, in a similar way, are associated to a identification key belonging to the same logical space and obtained,

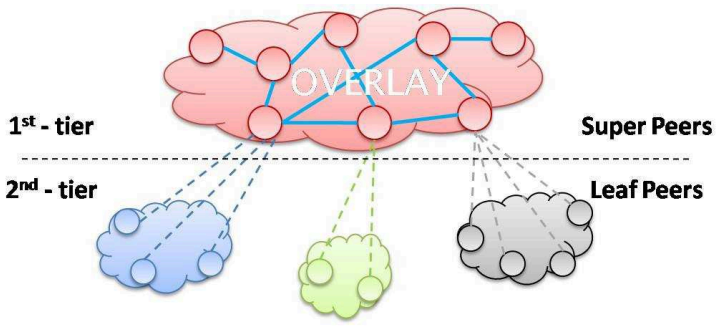

Figure 2-3: two-tiers hierarchical P2P architecture.

for example, hashing the title of the content. In a structured architecture each resource distributed in the network is managed by a unique responsible peer, usually the peer with the ID closest to the key value of the resource in the logical domain. In this way, when a node want discover or obtain a content it has to:

- hash the name (or other parameter, according to the employed scheme) of the requested content for obtaining the value of related key;
- send a lookup query for the key using the routing scheme adopted by the algorithm. Note that in this case the responsible peer is identifiable by the other peer and it is not necessary any form of flooding or random walk forwarding.

Note that the architecture needs only that peers hold a pointer to the resource they are responsible for; contents are usually stored in other locations. Thus, the reply to a lookup query is generally a pointer to the physical location where the resource is stored. Structured graphs obtained exploiting this approach enable efficient discovery of data items using the given keys. This feature is particularly useful when coped with large-scale networks and shared contents are not widespread. Although their efficiency in locating rare items since the key-based research/routing mechanism, structured algorithms incur in significantly overheads than unstructured ones for popular content. This is the reason

because over the internet today the unstructured P2P networks are more commonly used. The most important scheme belonging to this family are: Content Addressable Network (CAN), Tapestry, Chord, Pastry and Viceroy. We will discuss them in the following.

Another classification of the overlay structure can be made on the base of decentralization. We will say that a P2P architecture is **decentralized** whether the overlay system is distributed. If this is the case each peer participates to the building of the overlay network. Each peer, once received a lookup query, is able to forward it towards the correct destination comparing the value of the requested key with the entries of its routing table. Furthermore, resources are distributed uniformly among the peers and each of them is responsible for a well defined range of logical keys.

In **centralized** structure there is one or more servers the peers connect to in order to transmit and obtain information. Peers that want share and obtain contents have to contact servers for:

- publish the catalog of the resources they want to share with others members;
- receive information about the resources available within the network;
- receive the network address of peers holding the requested resources in order to download them.

Centralized systems suffer from a common disadvantage: the single point of failure due to the centralized search server. This

aspect obviously dramatically impacts on the system performance in terms of reliability and fault resiliency.

2.1 Unstructured P2P overlay networks

2.1.1 Napster

Napster [35] was the first commercial P2P system introduced in 1999. It pioneered the idea of a centralized overlay structure supporting file search facility. For the first time it introduced the concept at the base of modern file sharing: the distribution of popular contents need not to be sent to a central server from which users can get them; instead it could be handled in a more scalable and effective manner by many peers that have the requested contents. Following this approach users wanting to download data join the P2P network connecting to a central server. Once established the connection they can search if the desired file is available in the public catalog and, if this is the case, server will reply with the address of peers from which it is possible downloading the requested content. The experience of Napster, whose killer application was the mp3 files sharing, ended because of legal issues against the RIAA (Recording Industry Association of America). The lawsuit forced Napster to shut down the file-sharing service of digital music. However the new communication paradigm caught the imagination of platform providers and users

alike. Now Napster is again online and employs the original architecture design for commercial distribution of mp3 files.

2.1.2 Gnutella 1 e 2

Gnutella was the second major P2P system that appeared after Napster. It consists in a decentralized and unstructured architecture where peers joining the networks form a flat random graph. Each peer works as client and server at the same time. Like client it search for resources shared by others peers, like server it manages the lookup queries sent by the others nodes, answering with a lookup reply message if the researched content is available in the own catalog or forwarding the query message towards its neighbor peers if not.

To locate a data item a peer queries its neighbors and the typical query routing consists in flooding. Because of such design the protocol is extremely resilient to peers churning, i.e. peers joining and leaving the system frequently, and it present a high fault tolerance. But, on the other side, this search mechanism are not scalable generating high overhead when the number of peers grows.

When a node want to join the Gnutella network it has to connect to one of several well known host whose list is available online. Once connected the peer can send the messages to discover other participants and establish TCP connection with them. These

messages are simply flooded in the network. Thus, in such structure, the allowed messages are:

- **Group membership messages (PING-PONG)**: a peer joining the network send a broadcast PING message in order to notify its presence to the other node; the PING message is then forwarded by its neighbors and, at the same time, initiates a back-propagation of PONG messages. The latter are replies to the PING message containing information about the peer such as the IP address, the number of items it holds, etc. ;
- **Search messages (lookup query and reply)**: queries contain the references of the searched contents such as an identification string or a key value. Once received a query message the peer finds in its catalog and, if it is not the responsible peer for the resource, it broadcasts the lookup query towards its neighbors. When the request get the responsible peer it sends a lookup reply, back-propagated, containing the information necessary to download the content.
- **File transfer messages (GET and PUSH)**: when the lookup procedure ends, it is possible to establish a direct connection between the requester and the node holding the resource and file transfer can take place.

Scalability problems occur when the number of lookup or the number of peers increases. In such case the efficiency of the protocol is poor and for this reason it was be necessary introducing several improvements [51][52][53] to solve this issue.

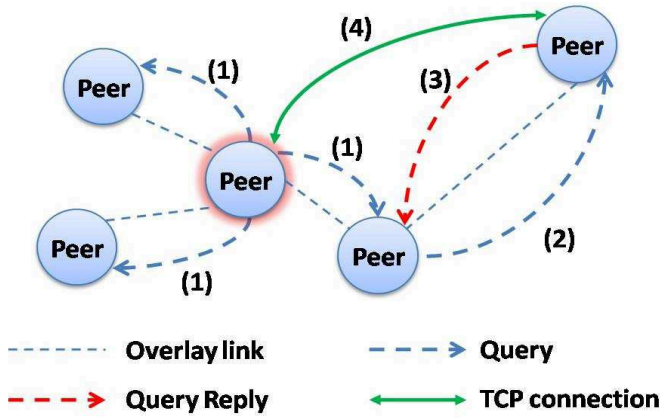

Figure 2-4: Gnutella architecture.

The latest versions of the algorithm, called Gnutella 2, uses the concept of hierarchical organization in order to minimize the number of queries flooded within the network. This approach is possible thanks the introduction of Hub or Super Peers [37], i.e. nodes characterized by high bandwidth connectivity, that play a key role in improving routing and information retrieval performances. Exploiting this network paradigm, only the super peer nodes concur in building and managing the overlay structure; they are, also, the only responsible of forwarding the query messages. When a peer, or leaf peer, want to join the network it has

to connect to a super peer and communicate its resources catalog. Likewise, when it want to search a content it sends the query message only to its responsible super peer that broadcast the query message towards its neighbor super peers.

This approach considerably reduces the traffic in the overlay and makes the system much more scalable compared to original Gnutella. Obviously, the complexity of Gnutella 2 is higher than that of the original system and the vulnerability of the system to attacks, resiliency, and fault tolerance increases as well because the super peers represent a single point of failure.

2.1.3 Torrent

BitTorrent is by far the most popular P2P protocol used over the Internet today. It is quite different from all of the systems considered above. It maintains a decentralized structure in resources distribution but, in order to obtain fast and fair download, it employs a centralized search mechanism. When a node want to retrieval a content it has to download the .torrent file making use of central-directory based search facilities; usually web sites serve for this purpose. The .torrent file contains information about the researched file/resource; in particular it reports the name of the file, its size and hashing information and, above all, the URL of the tracker. The latter keeps track of all the peers who have the resource (both partially and completely) and permits to establish

TCP connections among the peers interested in downloading the resource. To obtain load balancing in the P2P network, the tracker provides a random list of peers that have the file.

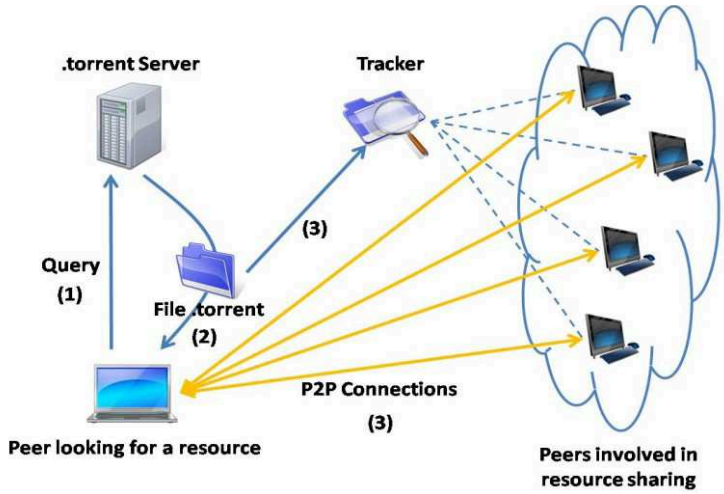

Figure 2-5: BitTorrent architecture.

With the aim of a fast spreading of contents the protocol cuts files into pieces of fixed size (256 Kbytes). Once established TCP connections among the nodes interested in resource sharing, each peer communicates the other with the list of pieces owned. At this point, the process of download/upload can begin. The general way for users is to download the rarest pieces of the file first, leaving the most popular ones for later. When a peer has the complete file it is called seed. In order to avoid the problem of free riders BitTorrent adopts a tit-for-tat policy according to which a peer responds with the same action that its other collaborating peer

performed previously. For the sake of simplicity, we can consider the case of a peer that downloading the pieces of a file does not contribute to the spreading of this resource reducing its upload rate: collaborating peers can use choking, that is a temporary refusal to upload towards the misbehaving node.

BitTorrent network is a very effective solution but suffers from legal issues (its killer application is the sharing of contents protected by copyright as video, music, software and games) and it shows several problems in finding and downloading old or unpopular contents because of the difficulties in creating the swarm of exchanging nodes interested in the sharing of these resources.

2.1.4 Edonkey

The eDonkey P2P system is one of the most successful and popular file sharing system. The Internet studies proposed in [69] show that the Internet traffic due to eDonkey is second only to BitTorrent protocol. Its most famous client is by far eMule, that have today more than 4 million users connected online.

The eDonkey search procedure follows the client/server architecture, where servers index the contents their clients provide. Note that servers only store information about the resources, but not the resources themselves.

The eDonkey network uses three types of communications:

1. **Server-server**: these communications performed over UDP connections are used by servers to maintain updated the list of other known servers and to forward the requests for contents (it is possible, in fact, to choice if a content research should be performed locally, i.e. among the peers managed by the server, or at whole network, i.e. extended to the peers managed by the other servers);
2. **Client-server**: performed via TCP connections are necessary for: (i) logging in a node that want join the network; (ii) communicating the catalog of resources the node want to share with the other peers; (iii) sending query messages to locate contents.
3. **Clint-client**: established to download/upload contents among peers.

When a node want to join the network it has to login onto a server via TCP by providing its IP address, the connection port and a list of files/resources it want to share. The list of servers, with the related addresses, is usually provided into the client or available via dedicated web sites. Once established the connection, server assigns a user ID to the node and update its local database adding the resources catalog of the latter. After this exchange of information the node is free to search and download files shared by other peers and, meanwhile, his files can be uploaded by other participants. The search procedure uses simple text search to locate files. As soon as the desired file is located user sends a query

sources message containing the hash (MD4 algorithm) of the file to its responsible server; the server answers with ID/port pairs of the clients that claim to store the file. At this point client-to-client connections start to transfer file. With the aim of making fast the retrieval procedure, eDonkey uses the multiple sources downloading strategy that consist in separating files in multiple pieces, called chunks, typically of 10 MB each, that can be transferred separately to parallelize the procedure downloading chunks from multiple peers simultaneously.

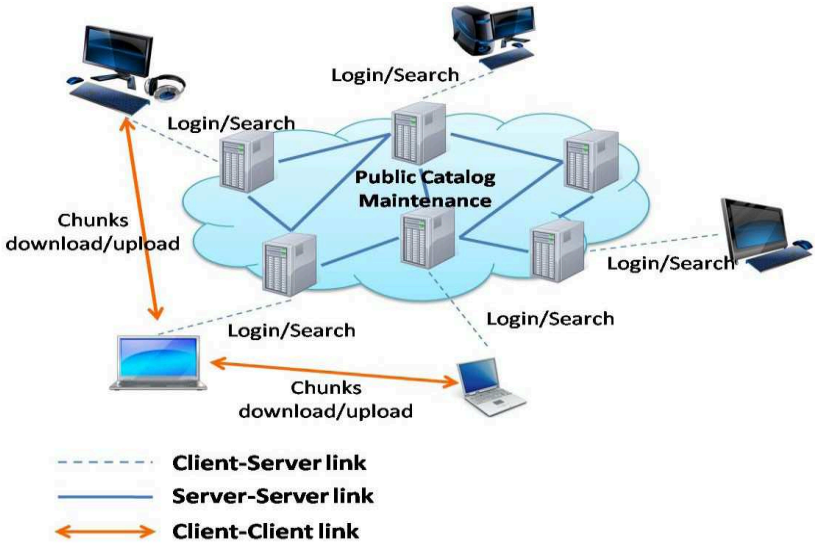

Figure 2-6: eDonkey P2P network.

The eDonkey protocol also avoids the problem of free riders by introducing a complex scoring mechanism that rewards collaborating nodes and penalizes peers active only in download.

The eDonkey protocol suffers from legal issues due to its principal use, the downloading of material protected by copyright; under this point of view its system of servers represents a single point of failure and it often happens that the related list must be updated because of the frequently interruption of the servers.

2.1.5 FastTrack

FastTrack belongs to the third generation of P2P networks. It employs an unstructured and decentralized approach. As well as Gnutella 2, in order to improve scalability, FastTrack exploits an hybrid architecture where we can distinguish two hierarchical level:

1. **Super peer nodes**, which are responsible for the building of the overlay network. They are responsible for a certain number of ordinary peers (the lower level of the hierarchy) they are connected to and they are the only appointed to perform the searching procedures within the network, generally by flooding;
2. **Ordinary nodes**, participate the network by connecting to a responsible super peer that index the resources they share and answer the search request on their behalf.

Super peers are nodes that possess high bandwidth, low latency, high computational and storing capacity. They are voluntarily elected to guarantee scalability and facilitate the search

procedures. The ordinary peers communicate the meta-data of the resources they are sharing to the responsible supeer peer; all the queries are also forwarded towards the responsible super peer. The choice of which super peer to connect is taken based on the workload of the super peers and locality considerations.

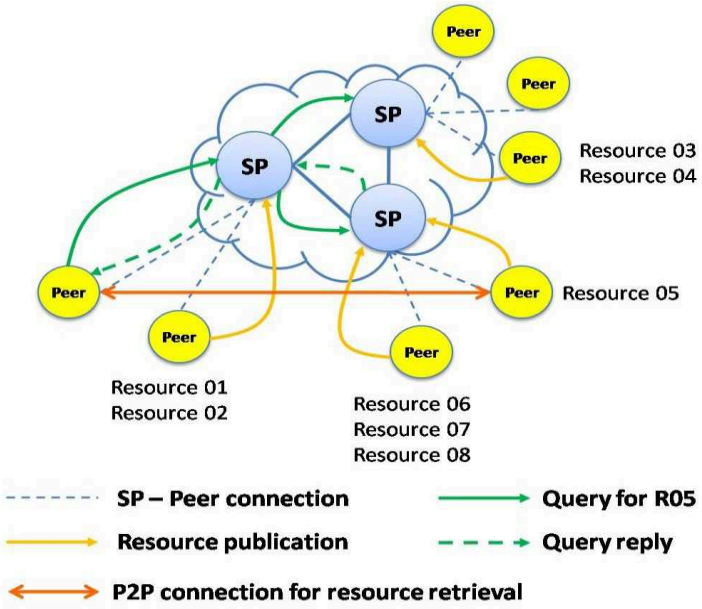

Figure 2-7: FastTrack architecture.

KaZaA is the name of the most famous FastTrack application. According to the assessments of [5] in 2008 there are approximately three million nodes in KaZaA network, of which 30.000 were super nodes. The list of available super peers are provided by a central server which can be considered a single point of failure in the system.

2.1.6 Freenet

Freenet is an adaptive P2P network of nodes that want to store and retrieve data contents identified by location independent keys. Each node maintains a local datastore which it makes available to the network for reading and writing operation, as well as a dynamic routing table containing the addresses of the other peers and the resource keys they claim to hold. The resources are randomly replicated among the peers participating to the network so Freenet is able to:

1. guarantee the information anonymity, because it is not possible to know the source for a certain content;
2. avoid the single point of failure issue, because each data object is connected to several responsible peers.

Routing procedure consists in passing the query messages along from peer to peer through a chain of requests in which each peer decides the next hop destination according to the requested key. Each request is associated a hop-to-live parameter, similar to the time-to-live used by the IP protocol, which is decremented at each node in the routing path in order to avoid infinite chains. Furthermore, a pseudo-random identifier is associated to each query message with the aim of avoiding loops in the routing procedure.

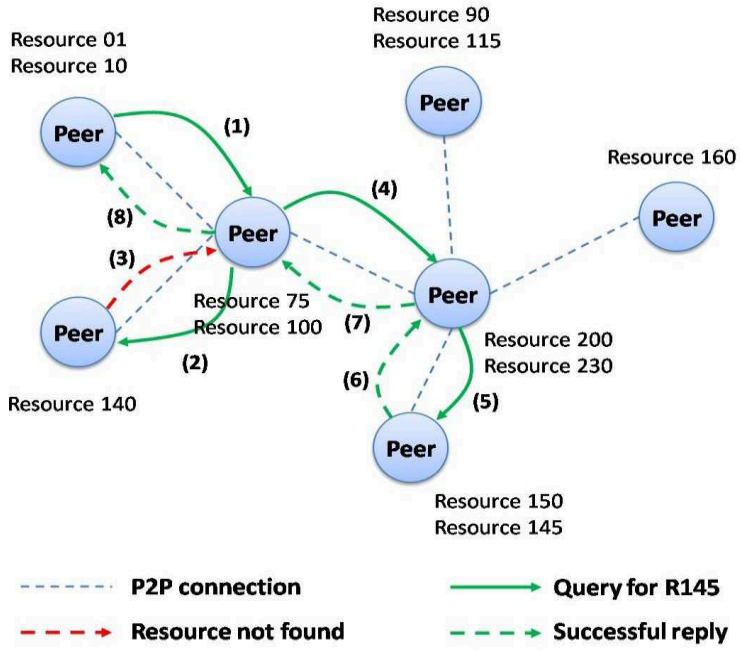

Figure 2-8: Freenet routing scheme.

The routing algorithm is designed to adaptively adjust routes over time and to provide efficient performance using peer's local knowledge. To retrieve a content, once calculated the related binary file key, a node send the query message towards the neighbor holding the nearest key to the requested one. If the latter is not able to retrieve the content then a new query message is sent towards the neighbor managing the second nearest key to the requested one, and so on. When the content is found, the node managing it will pass the data back to the upstream requestor and, at the same time, intermediate nodes will cache the file in their own local datastore creating a corresponding data entry in their routing tables.

Subsequent queries for the same key will be satisfied by the local caches.

Insert procedures are similar to the retrieval procedures. When a user want to publish a content it has to calculate the related binary file key and send a insert message towards the node belonging to the routing table holding the nearest key. When a peer receives the insert message, it check in its routing table if a keys collision occurred and, if not, it will forward the insert message towards the next hop peer according to the routing scheme. When the message reaches the last hop, defined by the maximum hop-to-live parameter, without detecting any keys collision, an "all clear" message will be propagated back to the publisher peer indicating that the insertion procedure was successful. Instead, if a key collision occurs the pre-existing content will be propagated back to the publisher node that will be forced to change the descriptive text string. Note that this procedure avoids the propagation of fake, indeed introducing a content having a false key implies the propagation of the original content.

Joining the network will rely on discovering the address of one or more of the existing peers and the file keys they manage in order to build the routing table.

Files in Freenet are identified by binary file keys obtained by using the 160-bit SHA-1 hash function. We can distinguish three types of file key used in Freenet:

1. Keyword-Signed Key (KSK);
2. Signed-Subspace Key (SSK);
3. Content-Hash Key (CHK).

Keyword-Signed Key (KSK). It is the simplest type of file key; it is derived from a short string used by the user to describe the published content, e.g. /games/fifa2012. The descriptive string is used as input to generate a public/private key pair: the public half is employed to obtain the file key by hashing; the private half is used to sign the resource for providing an integrity check that a retrieved data file matches its data file key. Using only KSKs it is not possible to avoid that two or more users choice the same descriptive string for publish different contents.

Signed-Subspace Key (SSK). It enables the creation of personal namespaces. The public namespace key and the descriptive string are hashed independently, XOR'ed together and then hashed to yield the data file key. To enable the resources retrieval each user publishes the descriptive string and the user subspace's public key. Storing data, instead, requires the private half of the key so, only the namespace's owner is authorized to publish contents.

Content-Hash Key (CHK). It is obtained from hashing directly the contents of the corresponding data file. In Freenet the CHKs are employed for creating and managing updatable contents and for splitting contents: using this approach each version or

section of a resource can be inserted in the common file system separately under a specific CHK.

2.2 Structured P2P overlay networks

This category includes algorithms characterized by a well defined structure in overlay building. Usually, peers and resources are mapped by hashing in a logical space and they are designed in order to assign a certain number of resources, uniformly distributed, to each peer of the network, that we will define the responsible peer for that resource. Note that with certain exceptions the responsible peer stores a pointer to the physical location of the content and not the resource itself. The structured graph at the base of overlay permits efficient discovery of data items using the given key also without any replication strategy. Thus, differently from unstructured approach, structured P2P network are particularly suitable for sharing contents and resources not widely spread within the network and their efficiency in retrieval this kind of resource justifies the overhead introduced for building, managing and maintaining the complex overlay architecture.

In its simple form this class of systems does not support complex queries: a lookup query for a desired content must report its exact key value and the consequent routing of the query message happens on the base of that value.

In the following we will present the most important structured protocols presented in literature.

2.2.1 Content Addressable Network (CAN)

The Content Addressable Network (CAN) [32] represents a DHT-based distributed decentralized architecture for P2P networks. It is designed to be scalable, fault-tolerant and self-organizing.

CAN employs a virtual multi-dimensional Cartesian coordinate space as its logical space. Each resource shared in the network is mapped, by a deterministic hashing function, in a point P of the multi-dimensional space. The hash function guarantees an uniform distribution of contents in the space. The entire coordinate space is dynamically partitioned among all the peers in the system such that every one of them possesses its individual zone within the overall space. Using this approach each peer is responsible for the resource whose key value P belongs to its region. In particular, the responsible peer store in its dataset the pair <key,value>, where key is the logical value corresponding to the P coordinate and value is the pointer of the location, i.e. the IP address, where the content is physically stored.

A CAN peer maintain a routing table with includes its neighbors peers: if we consider a d-dimensional Cartesian space the

number of entries in the routing table will be 2d, that is the addresses of successor and predecessor peers in the d coordinate.

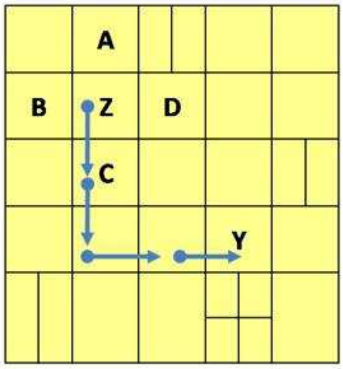

Figure 2-9: Routing in CAN.

When a peer want to retrieval a resource it has to obtain its key value by hashing its name; once obtained the logical identifier it can send a query message that will be routed along the peers composing the overlay. CAN's peers route the message towards its destination using a simple greedy forwarding to the neighbor peer that is closest to the destination coordinates. It is possible to demonstrate that CAN has routing performance of $O(d \cdot N^{1/d})$ against a size of routing tables equal to 2d, where d is the dimension of the logical space and N is the number of peers taking part in the network.

Such routing strategy requires a continuous coordinate space without unassigned regions. Thus when a node joins or leaves the overlay the space needs to be dynamically reallocated so that each point P has a corresponding responsible peer.

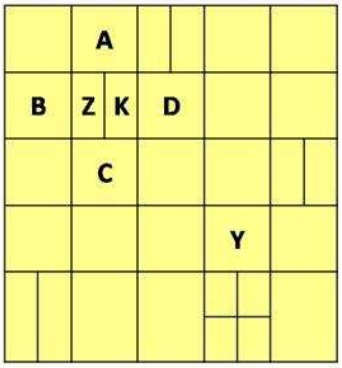

Figure 2-10: Region splitting in CAN.

When a new peer want to join the network a random destination is chosen and the region of the peer responsible for is subdivided into two parts. If we consider a two-dimensional X-Y space the region will be split, at the first time, along the X dimension and then along the Y dimension. Accordingly, the <key,value> pairs from the half zone to be handed over will be transferred to the new peer that, once obtained its zone, will learn the IP addresses of its neighbors and will give notice of its presence to them in order to update their routing tables.

In a similar manner the space should be reallocated when a peer leaves the network. In such case a takeover algorithm ensures that one of the leaving peer's neighbors takes over the zone. We note that the number of messages needed to react on a node joining or leaving the system is 2d.

2.2.2 Chord

Chord [33] is a DHT-based decentralized scheme based on a logical ring topology, called Chord ring. More in detail, peers and resources are mapped in the logical space by hashing their network address and name respectively. Usually, logical IDs and keys are m-bit identifiers obtained using SHA-1 as base hash function. Identifiers are, thus, ordered on a circle modulo 2m.

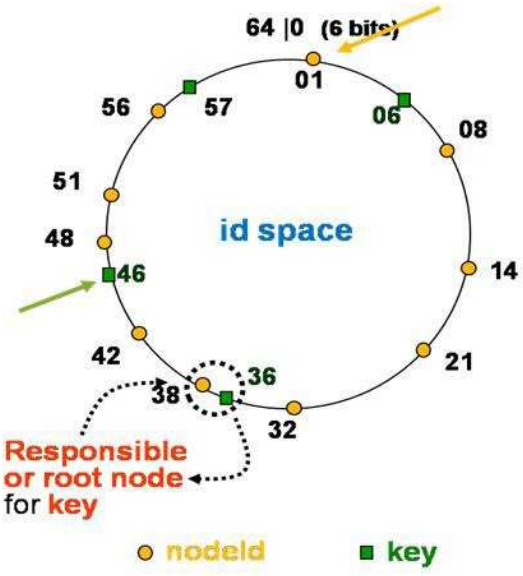

Figure 2-11: Chord logical ring

Using this logical space a resource having key value k is assigned to the peer whose logical ID follows k in the identifier space. In Figure 2-11 we show the case of a logical space obtained

using 6-bit SHA-1 hash function. The resource associated to the logical key 36 is assigned to the peer whose ID follows 36 in the Chord ring, i.e. peer identified by ID 38. The latter becomes the responsible peer or root node for resource 36.

For a resource k, thus, we can indicate successor(k) its responsible peer, that is the first peer clockwise from k in the logical ring. To maintain consistent hashing mapping the peers have to maintain updated their routing tables in case of nodes joining or leaving the network. In particular, when a node n joins the network certain resources managed by successor(n) need to be assigned to n. when a node leaves the network, instead, all of its resources are reassigned to successor(n). it is possible to demonstrate that the cost of peers joining/leaving the overlay is $O(\log N)^2$ where N is the number of peers in the network.

The lookup query message for a certain key is forwarded around the ring via successor neighbors until it reaches the responsible peer that manage the searched key. In Figure 2-12 is shown the lookup procedure initiated by the peer 8 and related to the research of resource whose key is 46. As represented, the lookup message is forwarded by each peer of the logical ring to its successor until the message is received by the peer 48 who is the responsible for the resource 46. Responsible peer sends a lookup reply to node 8 following the reverse of the path.

In order to build and manage the overlay structure each peer in the ring topology needs to know the address of its successor peer. But

this is not the only pointer towards other peers. Each node of Chord structure maintains a routing table with up to m entries, where m is the number of bits used to build the logical space, called finger table. If we consider a peer n, the i^{th} entry of its finger table contains the identity of the first peer that succeeds *n* by at least 2^{i-1} on the logical ring, i.e. i^{th} entry is successor(n + 2^{i-1}), where $1 \leq i \leq m$. Figure 2-13 shows the finger table of node 8. The table contains both the Chord logical IDs and IP address and port number of successor peers.

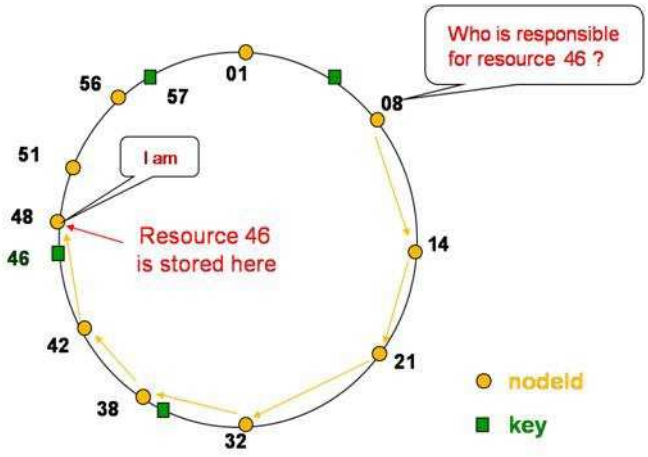

Figure 2-12: Lookup procedure in Chord.

When a peer joins or leaves the network the successors pointers of interested peers need to be changed in order to guarantee the correctness of lookup procedures. With this aim the Chord protocol uses a stabilization protocol running periodically in

background to maintain updated pointers in the finger table. To improve fault-tolerance each peer maintains a list of r successors rather than only one entry, in such way when the communication towards the successor fails it is always able to contact the next peer on its successor list achieving good reliability and fault resiliency.

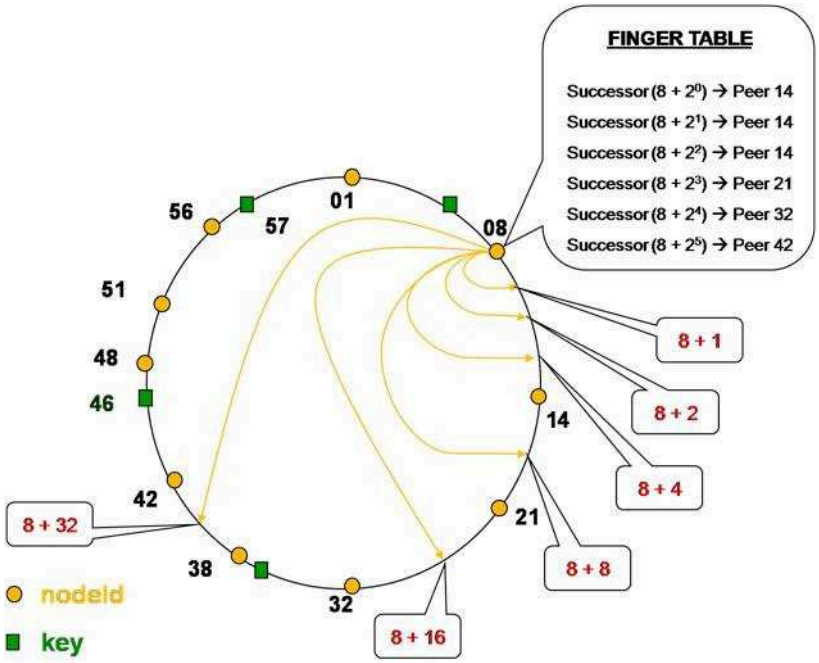

Figure 2-13: Chord - Construction of the Finger table.

2.2.3 PRR trees: Pastry and Tapestry.

The two algorithms described in this section are based on the Plaxton-Rajaraman-Richa trees [30]. In this kind of approach each node and each resource have assigned an ID in the logical address space in order to obtain an uniform distribution.

To find the owner of a resource whose ID is X, from a node with identifier Y, the algorithm lies in the following steps:

1. Let p = longest matching prefix between the IDs of resource X and node Y respectively;
2. Node Y finds in its routing table if it knows a peer whose ID matches Y for at least p+1 digits;
3. If the latter exists then Y forward the query message towards it;
4. Else the responsible peer for resource X is Y.

This represents a simplified exploitation of the mechanism but further details will be provided describing the two implementation, Pastry and Tapestry. Generally, routing is made possible by means of the construction of a routing table structured in different levels; if we consider the i^{th} level of the routing table of a peer, Y, this will contain the references of peers whose logical ID matches the Y's ID in the first I digit. In Figure 2-14 we represent the routing path of a query message created by node 3AF2 for the resource 47E2.

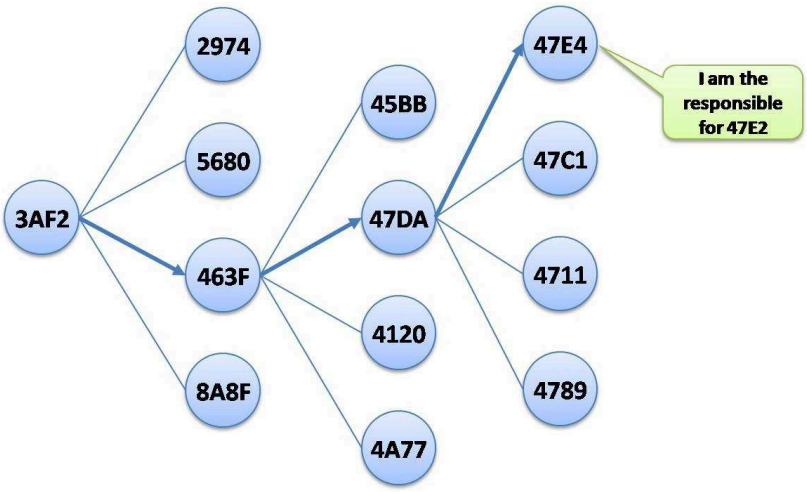

Figure 2-14: Plaxton-like prefix routing.

2.2.3.1 Pastry

Pastry [31] builds a self-organizing decentralized overlay network using an hybrid mechanism that makes use of:

1. Plaxton-like tree scheme (described above), for long range distance researches;
2. Chord-like ring scheme to improve routing exploiting the concept of logical proximity.

Each peer in Pastry is assigned a 128 bit identifier, the node ID. The logical ID is used to obtain the place of the peer in the circular Chord-like identifier space, which range from 0 to 2^{128} -1. The node ID is assigned randomly when peer joins the network and

it is assumed to be generated such that the resulting set of identifiers is uniformly distributed in the 128-bit space. Each Pastry peer maintains three list of peers:

1. Routing table;
2. Neighborhood set;
3. Leaf set.

Using the above mentioned information, Pastry is able to lookup a resource in less than $\log_B N$ steps, where N is the number of peers participating the network and B = 2b is a configuration parameter with typical value of b = 4. Under these assumptions, node IDs and resource Keys can be considered a sequence of digits with base B.

Now we can explain the structure of the three lists. A peer routing table is designed with $\log_B N$ rows, where each row contain at maximum B-1 number of entries. If we focus on the ith row of the table, the related B-1 entries refer to those peer whose node ID shares the current peer's node ID in the first I digits, but whose $(i+1)^{th}$ digit has one of the other B-1 possible values other than the $(i+1)^{th}$ digit of the current peer's ID. Note that if among these peers there are two or more with the same $(i+1)^{th}$ digit, the routing table will include the one closer according to the proximity metric adopted. We also observe that the choice of the parameter b is a trade-off between the size of the routing table, that is

approximately $(\log_B N) \cdot (B-1)$, and the maximum number of hops required to route between any pair of peers, that is $\log_B N$.

The neighborhood set contains the IDs of the M peers closest to the node; the concept of proximity used by Pastry is based on scalar proximity metrics such as the IP routing distance.

The leaf set makes use of the position of the node, X, within the logical ring; it includes L peers and, in particular, the L/2 nodes with ID smaller than X (i.e. the L/2 predecessors in the logical ring) and the L/2 nodes with ID larger than X (i.e. the L/2 successors in the logical ring). Typical values of M and L are B or 2B.

In Figure 2-15 we show a generic Pastry routing procedure to reach the responsible peer for a desired resource. Generally, we can describe Pastry routing as a three-phase routing:

1. the node check if the desired key falls in its leaf set and if the research succeeds it choose the peer numerically closest to the key;
2. if the lookup fails, it uses a Plaxton-approach forwarding the query message towards the appropriate PRR-style routing neighbor until the query message reaches a peer whose leaf set contains the key;
3. finally, it is chose the peer numerically closest to the resource key.

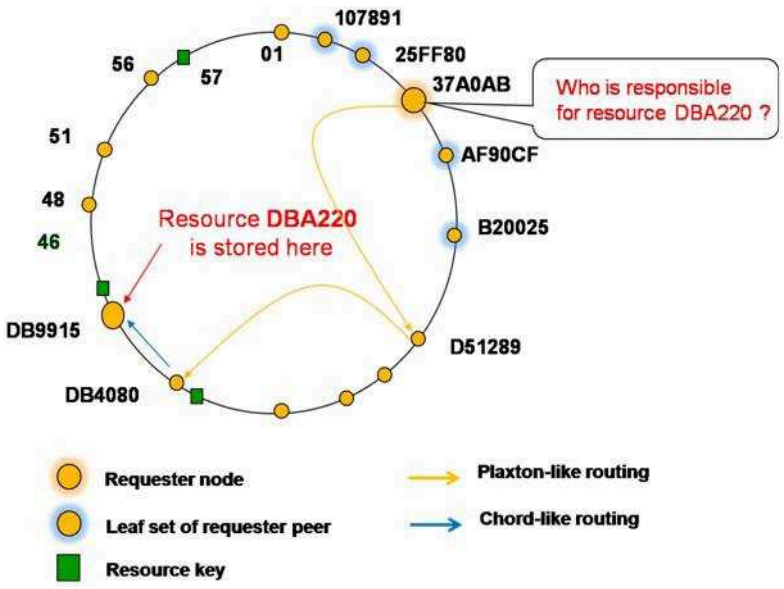

Figure 2-15: Routing in Pastry.

When a new peer, X, want to join the network it has to fill its routing table and leaf/neighborhood sets. This happen contacting a close peer, A, according to a proximity metric. Thus, X asks A to route a join message with the researched key equal to X. Let us suppose that peer Z is the destination of the join message, all the peers visited by this message will be informed of the presence of peer X and they will reply sending their state table. This ensures that X initializes its state with appropriate values and that the state in all other peers is updated. Similarly, when a peer X leaves the network, other peers have to update their state tables.

2.2.3.2 Tapestry

Tapestry [34] shares similar properties as Pastry: it employs decentralized randomness to achieve both load distribution and routing locality and uses Plaxton-like routing to route query messages towards the destination. Tapestry nodes do not make use of the leaf set lists as Pastry: Tapestry is thus a pure PRR-tree algorithm whereas Pastry represents an hybrid solution that combines PRR-tree and Chord-ring topologies.

This is not the only difference between the protocols. There are other two substantial differences:

1. the data object replication;
2. the handling of inexact matches.

Data object replication. If a resource ID is managed only by one owner, the responsible peer, the latter represents a single point of failure of the system. In order to improve the fault tolerance and resiliency of the system, Tapestry uses a sort of data replication thank to which each resource has multiple root or responsible peer managing the pointer to its location. When a content has to be published, Tapestry generates a well defined and constant sequence of "salt" values for each object ID, then hashes the results and starts the procedures necessary to find the root peers for each obtained key. In such a way, each resource is assigned to multiple root servers and the replication procedure, as well as improving the

fault tolerance of the overlay, improves the lookup procedure. When a peer want to start a retrieval procedure it can easily obtain the logical ID of the resource and the logical IDs of its related "salt" values; once calculated the IDs for the content multiple query messages can be sent towards the different destinations.

Handling of inexact matches. The second difference consists in the choice of the next hop peer in case of inexact matches. Pastry, exploiting the information contained in the leaf set, can forward the query and publish messages towards the peer whose logical ID is closest to that of the resource key. Tapestry does not manage this kind of information, thus it cannot know the exact place of each node in the logical space (Tapestry does not use the Chord-like topology as Pastry). The protocol solves this problem using the so called surrogate routing. This approach consists in choosing an object's root node to have the same logical identifier as its ID. Given the random nature of the logical addresses space domain, it is unlikely that this node exists. Even so, Tapestry operates as if the designated root node exists by attempting to route to it. A route to a non-existent identifier will encounter empty neighbor entries at various positions along the way. In these cases, the goal is to select an existing link which acts as an alternative to the desired link (i.e. the one associated with a digit of the researched key). This choice is done with a deterministic selection among existing neighbor pointers. Routing terminates when a

neighbor map is reached where the only non-empty entry belongs to the current node. That node is then designated as the surrogate root for the object.

Tapestry employs a fixed routing table with size equal to $B \cdot \log_B N$, where B is the base of the logical addressing space and N is the number of peers of the network. The routing performance is $O(\log_B N)$ and the cost of peers joining/leaving the network is equal to $\log_B N$ messages.

2.2.4 Bamboo

Bamboo [59] belongs to the third generation DHT solutions. It improves previous schemes such as Pastry by taking into account e.g. congestion arising due to large management traffic. While Bamboo is based on the routing logic of Pastry, management of overlay structure is different in order to be more scalable in dynamic environments.

To maintain the network structure, Bamboo uses two set of neighbor information at each node: leafset and routing table. The leafset consists of successors and predecessors that are the numerically closest in the key space. While two nodes may be neighbors (in the leafset) in the overlay, they may be physically far away. When doing a query, the latter is forwarded until a node which has the key in its leafset to ensure correct lookup is reached. However, using only the leafset during lookups results in

complexity of O(logN). To improve lookup performance, a routing table is used, which is populated with nodes that share a common prefix. Accordingly, routing table lookups are ordinary longest prefix matches. When data is stored in the DHT using the *put* command, the data is routed through the DHT to the node primarily responsible for storing the data. When the responsible node gets the data, it caches it within its leafset neighbors in each direction according to the number of desired replicas. For certain applications, the number of desired replicas can cause large demands for storage space. Therefore, for data storage updates, a node periodically picks a random node in its leafset and synchronizes the stored keys with it. The correspondent node calculates the set among its stored keys that should also be stored at the sender node, sending those keys to the sender, including the hash values of the data. The major difference between Pastry and Bamboo is the way they handle management traffic. In Pastry, management is initiated when a network change is detected, while in Bamboo management traffic is periodic regardless of network status. While reactions to changes in the routing layer operate on very small timescale, reactions to changes in overlay structure are not so fast. However, the approach to use periodic up-dates has shown to be beneficial during churn [59], since it does not cause management traffic bursts during congestion. Such traffic bursts can further increase packet loss probability, lead to management messages being dropped and other overlay network problems. In

order for Bamboo to be able to serve requests and maintain a consistent network view among its nodes, it needs to perform overlay maintenance message exchange between nodes. Periodic management traffic occurs at all layers of the Bamboo system. *Neighbor ping* is generated by every node in order to make sure that the node can still reach its one-hop neighbors in the overlay. It is also used to maintain a RTT estimate used for retransmission timeout calculations. Such timer values are used to derive, if e.g. members left the overlay or messages need to be re-transmitted. However, retransmitting too early will lead to too high number of packets. An accurate timeout value is crucial in order to predict if a packet is lost and needs to be resent along a different path in the overlay. Bamboo considers that two nodes share the same level when one node contains the other node in its routing table. Therefore, local routing table update is used to exchange the node information in that level. If a node gets information about other nodes that fit into the routing table, it probes the nodes to test reachability and to get a RTT estimate. If a node is reachable and fits into an empty field in the routing table, it is added. If the matching routing table entry is occupied, the node with the lowest latency is chosen. In standard configurations, Bamboo optimizes latency. It is important to note that an optimized routing table does not influence lookup correctness, but only lookup latency [60]. As wireless networks are rather limited in bandwidth, a balance between overlay lookup efficiency and management traffic

overhead is important [61]. The Bamboo system has been evaluated through simulation and using testbeds such as the PlanetLab [64].

2.2.5 Kademlia

Kademlia [38] is a decentralized P2P overlay scheme based on the XOR distance metric. Each peer and resource in Kademlia are assigned a logical ID in the 160-bit space. There is not only one responsible peer for a resource, but the pairs (key,value) are stored on the nodes whose IDs are close to the key according a distance metric based on XOR operation, that is $d(a,b) = a \oplus b = d(b,a)$.

To locate a key the Kademlia routing algorithm uses the same XOR metric to estimate the distance between the node ID and the key. The node forward the query towards those *m* nodes, belonging to its routing table, that are closest to the destination. The lookup procedure ends when the resource is retrieved and, additionally, Kademlia exploits the replication of the pairs (key, value) at the closest nodes that does not yet have the replica of the pair, improving the reliability of the system and alleviating hot spots along the lookup path.

The routing tables at each peer contain the triple (node ID, IP address, UDP port) and are organized in special lists called k-buckets. The parameter k depends on the implementation of the system and its value is usually equal to 20. Each k-bucket maintain

information about other peer of the network that are situated in a particular range of distance from the considered node. The k-buckets are updated during the routing processes and, more generally, whenever a node receives a message from another peer. Because the k-buckets have a limited size, the updating process is optimized to keep the longest living peers always in the routing table, as it has been shown that the longer the peer stay in the network the less probable it is that it will fail or leave in the future.

Kademlia requires $O(\log_B N) + c$ messages for the lookup routing, where c is a small constant. The routing table size is $(B \cdot \log_B N + B)$ and the number of update messages for nodes joining/leaving the network is $(\log_B N + c)$.

2.2.6 Viceroy

Viceroy [39] is a decentralized P2P network using the DHT to manage the distribution of data among a changing set of peers and to allow peers to retrieve any re source by name. Viceroy algorithm uses two type of logical topologies to route query message towards the destination:

1. **Chord logical ring**, where peers are placed according their logical ID obtained by hashing, for example, their IP addresses;

2. An approximation of the **butterfly network**, on the base of which peers are randomly distributed in different logical levels.

According to these two configurations, each peer maintain at maximum 7 outgoing links in its routing table:

1. **logical ring links**: they are two links towards the predecessor and successor in the logical Chord-like ring;
2. **level ring links**: they are two links towards the predecessor and successor in the same logical level of the butterfly topology;
3. **butterfly links**: they are three links connecting each peer with other peers belonging to different levels of the butterfly network; in particular:

    a. the **upward link** connects the peer P at level h (h>1) with the first (h-1)-level peer after P in the logical Chord ring;
    b. the **downward left link**, used for short range route, connects peer P at level h with the first (h+1)-level peer after P in the logical Chord ring;
    c. the **downward right link**, used for long range route, connects peer P at level h with the first (h+1)-level peer distant at least $1/2^h$ from P in the logical Chord ring.

In Figure 2-16 is represented a simplification of the butterfly topology employed by Viceroy: logical ring links are omitted and level ring links and upward links are shown only for level 4.

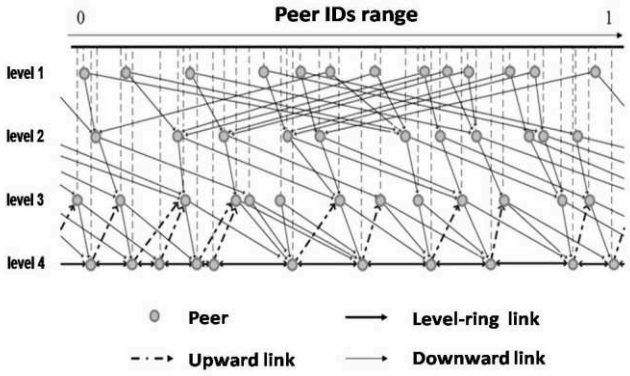

Figure 2-16: A simplification of the butterfly topology used by Viceroy.

The mapping of peers and resources is identical to Chord as Viceroy maps them to the unit ring. According to this a peer manages all key-value pairs whose value falls in the interval between its counter-clockwise neighbor's ID and its own ID.

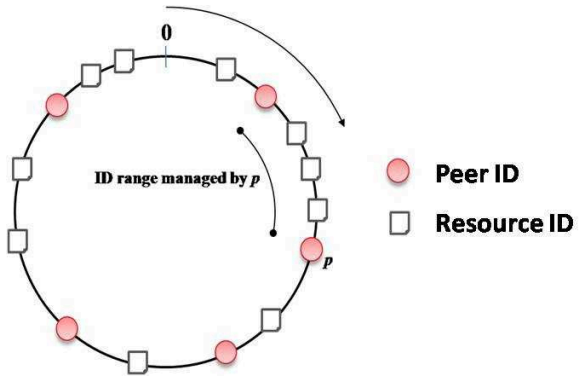

Figure 2-17: Chord-like mapping of peers and resources in Viceroy.

Exploiting both the butterfly and the Chord-like topology, information retrieval in Viceroy follows a three steps routing:

1. The query message is forwarded towards the level 1 of the butterfly by using the upward links;
2. Routing proceeds down the level of the tree according to the downward left or right links depending on whether the target key is at distance greater than $1/2^h$ or not; this continues until a node without downward links is reached;
3. Finally, a proximity routing is performed by exploiting the unit or level ring links.

It is possible to show that this process requires O(logN) steps where N is the number of peers in the network. The cost of peers joining/leaving the network is logN. Further details on Viceroy structure will be provided in Chapter 2 where we will discuss abut a location aware modification of Viceroy, called Georoy, whose application in wireless and delay tolerant environments represents the focus of this thesis.

2.3 Design guidelines in deploying P2P systems in wireless mesh networks

In this chapter we have given a brief description of the most popular overlay schemes for P2P networks proposed in literature. They have been proposed to address the requirements of large scale wired networks so, they do not represent efficient solutions for wireless environments. P2P architectures in wireless ad hoc and mesh networks are very challenging due to their dynamic, multi hop and often limited power/computational resources nature. Nevertheless, the above illustrated schemes represent an irreplaceable source of ideas in designing effective and suitable solutions for wireless networks and, with this aim, we have believed that a concise but at the same time exhaustive discussion about them was of primary importance. Extending the P2P paradigm to wireless mesh is fundamental because of two major reasons: mesh networking is a promising solution for the near future able to knock down the digital divide and spread the benefits of Internet access everywhere and anytime; P2P applications can play a key role in the diffusion of mesh paradigm meeting the growing need of communication and resource sharing among people.

While ad hoc and sensor networks represent very challenging environment due to, respectively, the high mobility and the power/computational constrains, for wireless mesh there are not so

many restrictions. Nevertheless, some considerations must be taken into consideration: the presence of fixed mesh routers with constant power supply allows the implementation of almost any type of service discovery architecture but, the mobile nature of mesh clients makes necessary adopting a **hierarchical structure**, including at least two tiers where mesh routers build the overlay network and are responsible for providing resources or contents on behalf of mesh clients that will represent the lower tier of the architecture.

Under the perspective of guarantying fault resiliency, scalability and flexibility to the network a **decentralized architecture** should be adopted instead of a centralized one.

Furthermore, P2P networks over WMNs should be able to handle a moderate level of churn, i.e. the portion of nodes in the networks that have a short lifetime and have a tendency for frequent and unpredictable failures or crashes. **DHT-based approaches** achieve good performance in information retrieval procedures and perform quite well under moderate level of churning, so they represent a viable solution for implementing P2P networks over WMNs.

In theory, most of the proposed DHT schemes can guarantee that any object in the network will be located in $O(\log N)$ steps. However, this number cannot be directly mapped to the number of physical hops, as the overlay topology often does not match the underlying physical topology. Maintaining a low number of

physical hops, under the perspective of applying the P2P scheme in a wireless mesh, could be a desirable condition. With this aim, introducing **location-awareness** in overlay construction represents a promising research direction in the field of wireless P2P systems. All the above mentioned considerations converge in the Georoy algorithm that represents the focus of this dissertation.

3 A location-aware P2P scheme for WMCNs: Georoy

Reliable and efficient P2P networking represents an interesting and attractive research area in the field of WMCNs' services. In this chapter we will describe a DHT-based algorithm, Georoy, proposed in [6], that represents a location-aware variant of the Viceroy scheme particularly suitable for application in wireless mesh networks. Furthermore we will present some improvements introduced to such protocol, as explained in [1], and finally we will evaluate the performance of the proposed overlay network by means of MatLab [65] simulations. A more exhaustive simulation study, performed using Network Simulator 2 [62], will be presented in the next chapter.

3.1 Algorithm overview

In Georoy, differently from Viceroy, peers are organized according to a hierarchical architecture where we can distinguish two tier: the lower tier is made of wireless mobile leaf peers (LPs), which are content providers, and the higher tier is composed of

super peers (SPs), typically mesh routers implementing a distributed index of the available resources. Only SPs take part in the overlay construction. Resource keys associated to contents are assigned based on a predefined hash function known to all network nodes in the range [0, 1]. SP IDs are mapped in the same ID space of resources, i.e. [0,1]. So, both the SPs ID space and the resource keys space are mapped in the same interval [0, 1], building a Chord-like logical ring where each key, i.e. each resource shared by the leaf peers, is managed by the super peer with the smallest ID larger than the key ID as shown in Figure 3-1. The unit ring is composed of all SP nodes connected sequentially based on their IDs.

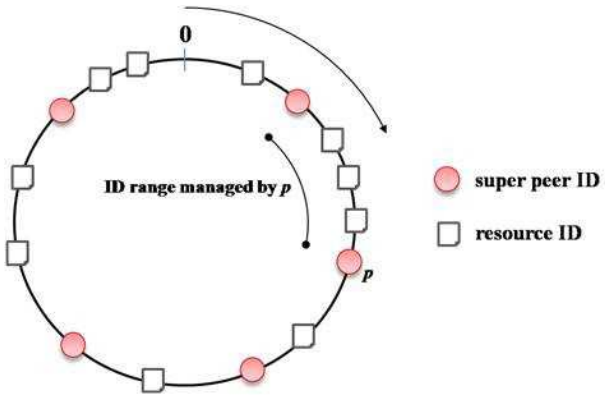

Figure 3-1: Chord-like logical ring used by Georoy.

As an example if the network is composed of only 5 SPs whose IDs are 0.2, 0.3, 1, 0.78 and 0.9, respectively, it follows that the unit ring is 0.2, 0.3, 0.78, 0.9 and 1. The ring is cyclic, which

means that node 1 is the predecessor of node 0.2 and, similarly, node 0.2 is the successor of node 1.

In the considered overlay, like in Viceroy, also short and long range links are considered between peers and are obtained by combining the unit ring topology with an approximation of the butterfly network that is a multi-stage network, where a node at stage i is connected with a limited number of nodes at stages (i-1) and (i+1) (as shown in Figure 3-2).

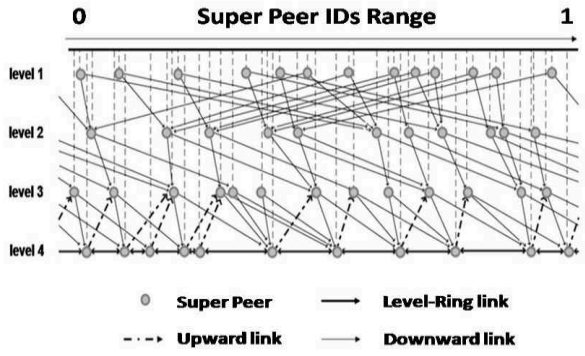

Figure 3-2: Butterfly topology in Georoy.

To set up the butterfly, each peer is assigned a certain level in the network. The identity of a node in the network is thus composed of the pair (p; l_p), where p is the peer ID and l_p is the level of node p in the butterfly. The content search procedure initiated by a source LP relies on exploration of the butterfly looking for the ID of the SP who knows the location of the LP providing the searched resource.

Algorithm overview

The exploitation of a hierarchical structure of the P2P network is not the only novelty of Georoy compared with Viceroy. The major goal of Georoy consists in improving the matching between logical and physical network. This aspect is made possible assigning SP IDs so that geographical proximity is respected, i.e. two nodes which are geographically close should be assigned close IDs in the logical addresses space. In this way it is possible to achieve a close correspondence between the physical and the logical topology. The mapping function that makes this possible will be discussed in the following section of the chapter.

Besides the above cited aspects, in [1] we introduce two important features in Georoy:

1. Epidemical resource replication, obtained exploiting leaf peers' mobility in order to both improve the resource availability and speed up the information retrieval procedure;
2. Trust preservation, obtained exploiting a blacklist where the super peers update the information about well behaving and misbehaving SPs based on the reliability of the contents they share.

In the following sections we will describe the Georoy's procedures needed to build and maintain the overlay network; furthermore we will present the procedures related to leaf pears' management. Finally, in the performance evaluation section we

will provide the reader with a simulation study of Georoy when the resource replication mechanism and the blacklist are used.

3.2 Georoy's Overlay Management procedures

3.2.1 ID and level assignment

Super peers are assumed to be distributed in a square region R of side s, for some constant $s > 0$ and are assumed to be aware of their position.

When joining the network for the first time, a SP p takes an ID in the interval [0; 1] which is related to its position. A mapping function is considered which is used to derive the ID. The mapping function, introduced in [6], permits to relate SP coordinates to values in the interval [0, 1] and is defined as follows:

$$M(x,y) = \begin{cases} \frac{x \cdot \Delta}{s^2} + \left\lfloor \frac{y}{\Delta} \right\rfloor \cdot \frac{\Delta}{s} & \text{if } \left\lfloor \frac{y}{\Delta} \right\rfloor \text{ is even} \\ \frac{(s-x) \cdot \Delta}{s^2} + \left\lfloor \frac{y}{\Delta} \right\rfloor \cdot \frac{\Delta}{s} & \text{if } \left\lfloor \frac{y}{\Delta} \right\rfloor \text{ is odd} \end{cases}$$

where (x,y) denotes the coordinates of peer p in the geographical area R and Δ is an arbitrary constant with $0 < \Delta < s$. This function preserves logical and physical proximity and builds a sort of "snake" traversing the interest area as shown in Figure 3-3. To this

purpose, the considered region R is divided into s sub-regions of equal area. All the nodes in the same sub-region are mapped into the same segment of the unit ring, where the position of the node within the segment is determined by its coordinates.

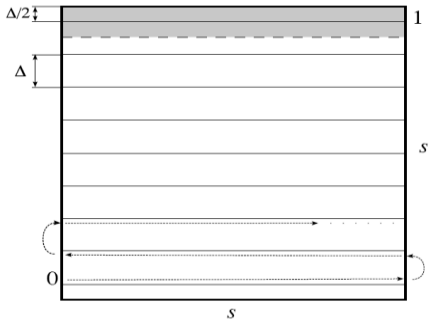

Figure 3-3: Division of the geographical area in sub-regions for Georoy's mapping.

This choice of the ID mapping function is different from the one proposed in Viceroy and is useful in that it allows to relate the physical location of two peers to the logical proximity of their keys so that logical and physical searches are closely mapped. Once a SP got its ID according to the methodology discussed above, it randomly chooses its level l_p in the butterfly by invoking a *LOOKUP procedure* as described in detail in the following.

3.2.2 Overlay construction and routing

To set up the overlay, 3 types of directed links are assumed:

1) **unit-ring links**, which connect a peer with its predecessor and its successor in the unit ring;
2) **level-ring links**, which are used to form a virtual bidirectional ring between the peers at the same level;
3) **butterfly links**, composed of an upward and two downward links. The **upward link** connects peer *p* at level h > 1 to the first (h-1)-level peer after position *p* on the unit ring. The **downward left link** (the short range link) connects *p* to the first (h+1)-level peer after position *p* on the unit ring; the **downward right link** (i.e., the long range link) connects *p* to the first (h+1)-level peer after position $(p + 1/2^h)$ on the unit ring.

So, every peer in the Georoy overlay network has no more than 7 outgoing links: 2 unit-ring links, 2 level-ring links, and 3 butterfly links.

Routing in Georoy is essentially performed by invoking a *LOOKUP(x; y)* function, where x is the requested key and y is the ID of the peer that invoked the function. Please observe that a search procedure is initiated by a LP by contacting the SP in its closest proximity. However, the LOOKUP procedure is initiated by

the SP. So the ID y refers to the SP initiating the search procedure, not the LP. This SP *y* will store a table mapping the leaf peer ID to the searched resource x so that it is able to map the contents being searched to the searching LP. The result of a *LOOKUP(x; y)*, once reached the responsible super peer which manages the key *x*, is the ID of the SP where the resource is physically held.

The LOOKUP(x,y) query is routed in the overlay network following a three-steps process:

1) **up to the root**: starting from node *y*, the query message is recursively forwarded upward in the butterfly, using the upward link, until level 1 is reached;
2) **traverse the tree**: the request is forwarded downward in the butterfly, using either the short or the long range link depending on whether *x* is at distance smaller than $1/2^h$ from the current position or not;
3) **traverse the ring**: finally, when the current peer has no downward links or it overshoots the target key, the lookup is forwarded exploiting the level-ring or unit-ring links until the responsible super peer managing key *x* is reached.

3.2.3 Overlay maintenance

A super peer y in the overlay maintains the following information:

1) its ID on the unit ring, y;
2) the current level l_y;
3) the connections on the unit ring, $pred_y$ and $succ_y$;
4) the connections on the level ring, $pred^l_y$ and $succ^l_y$;
5) the upward butterfly connection, $succ^{(l-1)}_y$;
6) the downward butterfly connections, $short^{(l+1)}_y$ and $long^{(l+1)}_y$;
7) the ID of the responsible peer for each key it is the Home SP of (the explanation of the term Home SP will be provided in the following sections);
8) a black list storing the IDs of all the SP it recently received bad/damaged/unreliable files from. The introduction of a black list is a novelty with respect to Georoy and plays a key role in order to preserve trustworthiness.

When a new peer y joins the network, it first selects its ID as described above. By invoking *LOOKUP(y;y)*, node y finds its successor $succ_y$ in the ring, and establishes a connection to it. Through it, peer y knows its predecessor $pred_y$ in the ring. Then, $succ_y$ transfers to y all the key-value pairs whose key is between

$pred_y$ and y. After that, peer y selects the current level l_y in the butterfly and finds its successor $succ^l_y$ and predecessor $pred^l_y$ in the level ring. Finally, node y establishes the butterfly links by finding $succ^{(l-1)}_y$, $short^{(l+1)}_y$, and $long^{(l+1)}_y$.

When peer y leaves the network, it has to remove all its established connections, notifying all neighbors in the overlay to update their links; then y transfers its contents to its successor in the unit ring.

When the current level changes, peer y has to update its level ring connections and butterfly connections, notifying the neighbors when necessary.

3.3 LP nodes' management procedures

3.3.1 Joining/leaving procedures

A LP u, upon entering the network, needs to invoke a join procedure to register its available catalog of resources. Accordingly, listening on the wireless interface u selects the SP with the best received quality which we denote as responsible SP $p(u)$, and registers by providing it with the list of the resources it is willing to share. Such information is maintained up-to-date by $p(u)$ in a local database of available resources. Let's call U_p the set of LPs $p(u)$ is responsible for. Also, we assume that for each LP resource, v, there is a Home SP, $p(H)(v)$, which does not change as

the LP storing v moves. Observe that, in general, $p(H)(v) \neq p(u)$. The Home SP of v is responsible for maintaining up-to-date the information about the location of v, i.e. its current responsible SP, $p(u)$. Figure 3-4 clarifies the different kind of nodes in the P2P network (leaf peers, responsible super peers and resource home super peers) and the roles they play.

Node u stores another list with the resources it was willing to share the last time it was connected to the network. If there are changes, then it must inform the Home SPs $p^{(H)}(v)$ $\forall$ v owned by u. When a LP u leaves the community network, the list of resources available in the network has to be updated. To this purpose, node u notifies its responsible SP, $p(u)$, before leaving the network, and $p(u)$ puts the resources shared by u in park mode in its local database. Meanwhile, $p(u)$ should provide a strategy to replicate the resources of u as discussed in detail in the following. If replication does not succeed, the Home SP, $p^{(H)}(v)$, must be informed that u is leaving the network so that its resource v should be put to park mode. If the replication succeeds, instead, the Home SP $p^{(H)}(v)$ should not be updated since at least one copy of v is still available and node $p(u)$ is still the responsible peer for it.

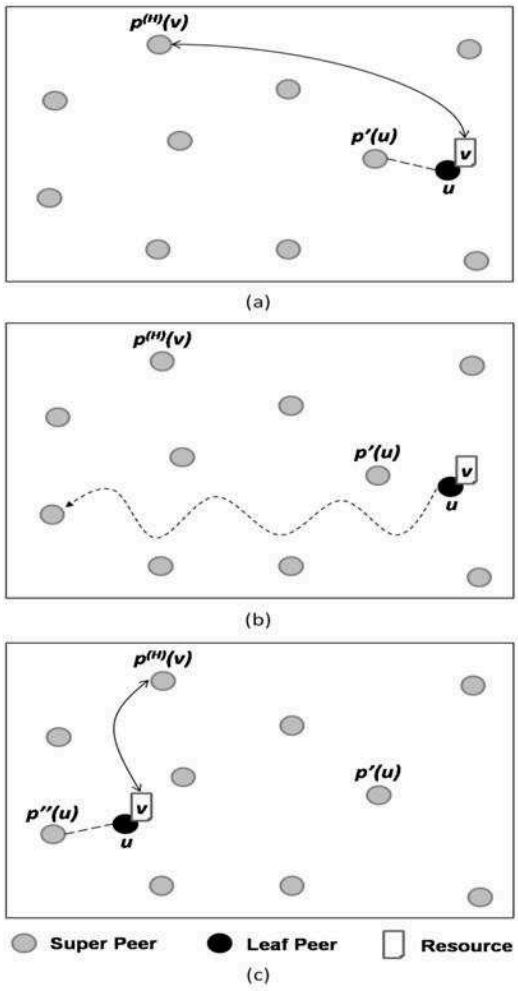

Figure 3-4: Overview of the P2P network. Leaf peer *u*, listening on the wireless interface, selects the SP with the best received quality, *p'(u)*, as its responsible super peer. Each resource *u* will share is managed by a home super peer according to the procedure discussed in paragraph . when *u* moves from its location and enters the coverage area of another super peer, *p''(u)*, the latter becomes its responsible super peer and $p^{(H)}(v)$ has to update the pointer to resource v accordingly.

If *u* joins again the same SP within a given time lower than a cancellation timeout, the only operation required is to put the resources shared by *u* in the available mode through a de-tagging in *p(u)*; also the catalogs of the resources stored by $p^{(H)}(v)$ ∀ *v* owned by *u* should be updated. In this way signaling in the network is reduced at a minimum.

Note that a LP *u* can also detach from the network without notifying its responsible SP in case of failure. Accordingly, the LP sends every T_{up}, an OK-message to its responsible SP. If, after T_{up}, the SP *p(u)* does not receive any message from *u*, then it sends a beacon to *u*. If *p(u)* does not receive any answer from *u*, it labels the resources shared by *u* as in park mode and informs the Home SPs *p(H)(v)* ∀ *v* owned by *u*. Please consider that, if the LP does not notify the responsible peer about its detachment, the responsible peer cannot provide a procedure to replicate *u*'s resources.

3.3.2 Insertion/removal of shared resources to/from the distributed catalog

Suppose that LP *u* wants to share a new resource *v* in the community. In this case, node *u* informs node *p(u)*. This node evaluates the key identifying the new resource through hashing, inserts the new resource in the catalog of the locally available resources, and forwards all information required to localize *v* to the

SP which is responsible for managing the corresponding key range, $p^{(H)}(v)$. Observe that if $p(u) = p^{(H)}(u)$ no update should be sent to the Home SP. Similar procedures are executed when a node does not want to share a certain resource anymore.

3.3.3 Resources replication

Procedures needed to keep updated the network about availability of multiple copies of the same resource should be considered. Availability of multiple copies of the resource can happen in the following cases:

- another copy of the resource v is inserted into the network. As an example, let us suppose that a new song by Madonna is published. It is expected that many users will download it through mobile phones or laptops and that this song will become very popular in few days. This means that multiple copies of the resource are expected to be progressively available into the network owned by many LPs. If this is the case, the joining procedure should take into account indexing of the same resource at many LPs.
- If a LP node migrates into the network, its catalogue of resources could be inherited by LPs currently located in the proximity of the previous LP location, exploiting

nodes' mobility to disseminate multiple copies of the available resources.

In the first case the v's Home SP, $p^{(H)}(v)$, is informed by the current responsible peer $p(u)$ about the availability of another resource with a given ID v. Accordingly the Home SP, $p^{(H)}(v)$, will have a table with multiple entries where for each resource v the responsible peers are listed. As an example, let us suppose that resource labeled as $v = 0.98$ is available in three copies in the network. This means that resource v can be located at three different LPs. Let's call α, β and γ these LPs. Each one will have its responsible SP according to their respective location. Let's call $p(\alpha)$, $p(\beta)$ and $p(\gamma)$ the responsible peers, respectively. If the Home SP for resource v is $p^{(H)}(v)$, this node will store a table saying that resource v can be located at nodes $p(\alpha)$, $p(\beta)$ and $p(\gamma)$. Upon receiving a $LOOKUP(v;w)$ issued by a LP whose responsible peer is node w, $p^{(H)}(v)$ will answer by providing w with the ID of one responsible peer among $p(\alpha)$, $p(\beta)$ and $p(\gamma)$ which minimizes the distance. Due to the mapping being chosen which preserves both logical and physical proximity, the Home SP will provide φ, subject to $\varphi = min_{i \in \{p(\alpha), p(\beta), p(\gamma)\}} |w - i|$. In this way the overall searching and retrieval redundancy is fairly distributed in the network, thus also decreasing the notification delivery delay.

Another situation which causes an increase in the number of copies of a given resource v currently available in the network is

related to LP migration. More specifically, when a LP moves into the network, this can be used to foster proliferation of a resource which is highly requested and downloaded. To this purpose, as soon as the LP *u* notifies its responsible peer about its will to leave, before leaving the network, the responsible peer *p(u)*, with a given probability which is a function of the popularity of a file, downloads the resources and distributes them among the LPs currently available in its domain. In this way the following advantages are met:

- the higher is the popularity of a file, i.e. the rate of requests for a certain resource, the higher is the number of copies of this file that can be located in the network;
- the Home SP for a certain resource should not be updated until the LP reappears in the coverage area of a new responsible SP (when the LP joins another SP, the Home SP should be informed about a new copy of the resource being available);
- to avoid excessive redundancy of copies in the network, only highly requested resources are replicated and this replication does not happen with probability 1 but with a random probability which depends on the degree of popularity.

3.3.4 LPs handover

Suppose that a certain LP *u*, which was formerly associated with SP *p'*, migrates into the coverage area of another SP, *p''*. In this case the following operations are required:

1) informing node $p^{(H)}(v)$ that from now on $p(u) = p''$;
2) updating if necessary (i.e. if the replication process has lead to multiple copies of a resource) the catalog of the resources stored by *u* from the catalog of the resources locally available at *p'*;
3) inserting the resources stored by *u* into the catalog of the resources locally available at *p''*.

Observe that the main operation involved is issuing local signaling between the leaf peer *u* and the old and current responsible SPs, *p'* and *p''*, to notify the change of location of node *u*. Then the old responsible peer *p'* could duplicate the resources initially stored by node *u* with a certain probability and the Home SP, $p^{(H)}(v)$, should be updated so that it learns the availability of new copies of *u*'s resources.

3.3.5 Information retrieval

Search requests are issued at the lower tier, and are routed in the overlay at the higher tier. When a LP *u* issues a request for a

certain resource v, it forwards such request to its responsible SP $p(u)$. The SP $p(u)$ first checks whether the request can be satisfied locally using the local available resources database. If the request cannot be satisfied locally, that is the corresponding resource is not stored by any of the LPs belonging to U_p, node $p(u)$ initiates a search in the overlay network. The result of the search algorithm will be the identifier of the Home SP of v, that is $p^{(H)}(v)$. The request will thus be forwarded towards $p^{(H)}(v)$, which stores information about the current location of v. As discussed above, if the Home SP $p^{(H)}(v)$ knows that multiple copies of the resource v are available in the network, it estimates the minimum distance between node u and the set of responsible SPs and, if the resource is available, it sends back the identity of this SP to u. If the resource is currently in park mode since there are no other copies available, the SP $p^{(H)}(v)$ informs node u that the requested resource is currently not available and u can decide whether to wait until the resource becomes available or not. If the LP which owns v moves or switches off during the resource transfer, the download can still go on by rerouting the transfer to any other LP which can provide the resource. Finally, if the requesting LP, u, moves or switches off during the resource transfer, the download stops and is restored when u becomes available again.

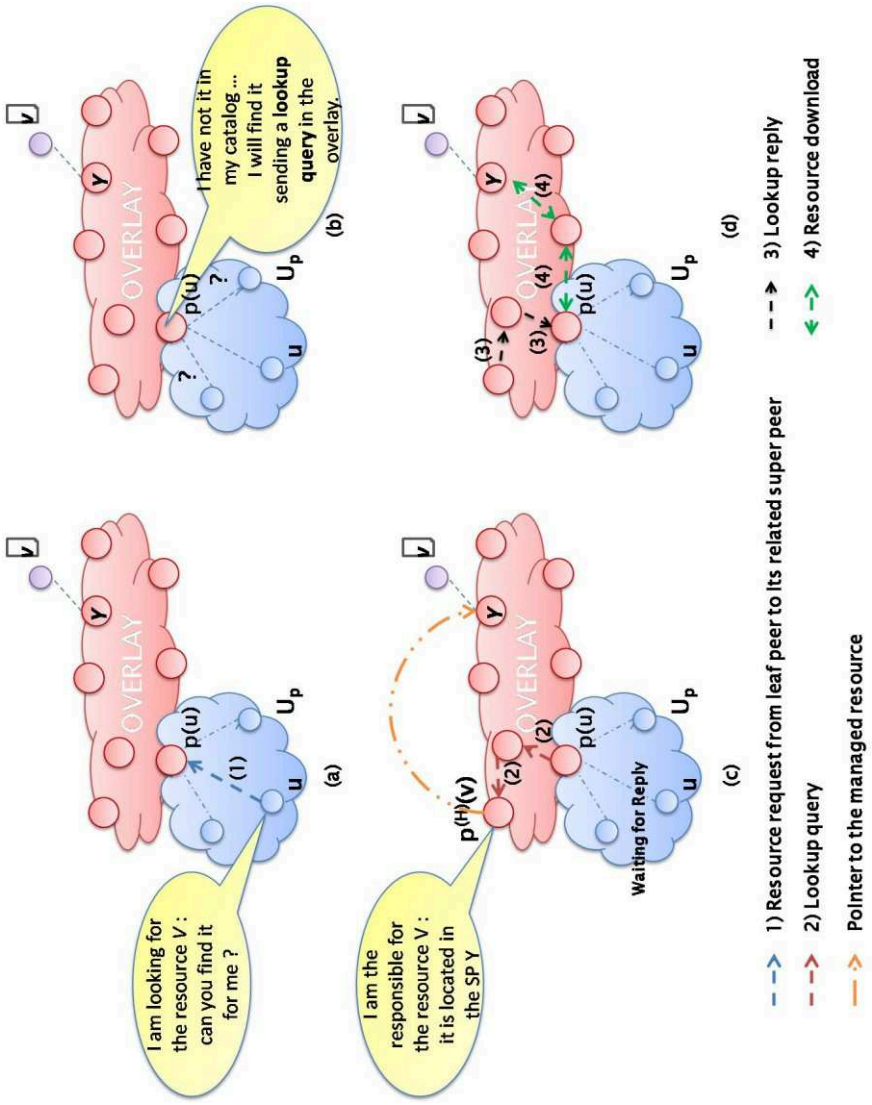

Figure 3-5: Information retrieval procedure in Georoy.

3.3.6 Trust preservation

In order to avoid trust problems, which are typical in common P2P scenarios, it is needed to keep updated the information about well behaving nodes. More specifically, we assume that each Home SP owns a blacklist of misbehaving or damaged SPs, i.e., nodes which have recently provided incorrect copies of the requested resources. This blacklist can be used to trace and punish SPs which provide unreliable contents to searching LPs. More in depth, this blacklist can be used to drop the search requests issued by misbehaving nodes in case these nodes want to access the network and perform resource requests. Let's briefly illustrate the procedures of denial of retrieval and alternative re-routing.

As regards denial of retrieval, the Home SP for resource *k* upon receiving a LOOKUP request issued by a SP that is inserted in its blacklist due to its reiterated misbehavior, notifies this SP about unavailability of the requested resource which, thus, will not be retrieved even if available in the network. Here we are assuming that if a SP is considered as misbehaving by at least one SP, it will be estimated as malicious also by the other SPs. This can be achieved by periodically exchanging blacklist data and updates between SP nodes.

Alternative re-routing is invoked in the following case. Let us assume that a Home SP storing information about the location of the leaf peers which possess resource *k*, once received a LOOKUP

request for *k*, and knowing that *k* is stored by many LPs which refer to different responsible SPs, checks its blacklist and bypasses the entries related to misbehaving nodes. So the Home SP re-routes the request to access the resource *k* from another trusted SP node, although farther away. The selected SP will be the one exhibiting minimum distance from the searching SP, apart from the malicious ones. This obviously implies a reduction in the efficiency of the search procedure in spite of trust preservation.

3.4 Performance evaluation of Georoy

In order to test the performance of the system, in case of multiple copies of the resources available in the network, we performed Matlab simulations. To this purpose we considered a wireless mesh network organized in two tiers consisting of N_{SP} super peer nodes and N_{LP} leaf peer nodes, with N_{SP} = [16; 25; 36; 49; 64; 81; 100; 121; 144], and N_{LP} = 1000, respectively.

The network area is supposed to be a square 1000x1000 m^2 large. Furthermore, LP nodes are assumed to be distributed randomly in the considered area, i.e. their coordinates are chosen randomly in the allowed region. Two configurations of the SP nodes were considered: nodes distributed over a grid or randomly.

Furthermore, we assumed that in the case of a grid topology SPs p_i and p_j , with i, j ∈ [1;N_{SP}] and i ≠ j, located in the positions

(x_i; y_i) and (x_j ; y_j), respectively, are connected by a link if and only if one of the four following conditions holds: (a) $x_i = x_j$ and $y_i = y_j + \Delta$; (b) $x_i = x_j$ and $y_i = y_j - \Delta$; (c) $x_i = x_j + \Delta$ and $y_i = y_j$; (d) $x_i = x_j - \Delta$ and $y_i = y_j$ where Δ is the size of one of the stripes identified in the network area. In case of a random topology, similarly, we assumed that node *i* and *j* are connected if and only if $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \Delta$.

Results have been obtained by averaging each simulation 100 times. In order to test the performance of the Georoy algorithm when multiple copies of the same resource are available in the network, we considered the mean number of physical and logical hops supported by the two topologies. This is shown in Figure 3-6-9. As expected, the use of an increasing number of copies disseminated in the network allows to progressively reduce the average number of both physical and logical links. Accordingly we observe that, when a sufficient number of replicas is available, the physical and logical searches exhibit almost the same behavior. When few replicas are available, instead, the use of a grid topology allows to cover almost uniformly the entire network while the random topology does not, so leading to better performance in terms of minimization in the number of hops.

Concerning the logical hops, as expected, there is no big difference in the two approaches since the logical overlay search is independent of the physical node location.

So an interesting insight of this replication mechanism is that the higher the number of copies replicated, the higher the search efficiency and thus the lower the number of physical hops traversed by a request. However, having too many copies of the same resource implies introducing some overhead which should be avoided. Accordingly, the provided plots allow to observe that there is a threshold value in the number of copies so that no significant improvement in the search efficiency can be achieved even if adding more copies of the resource.

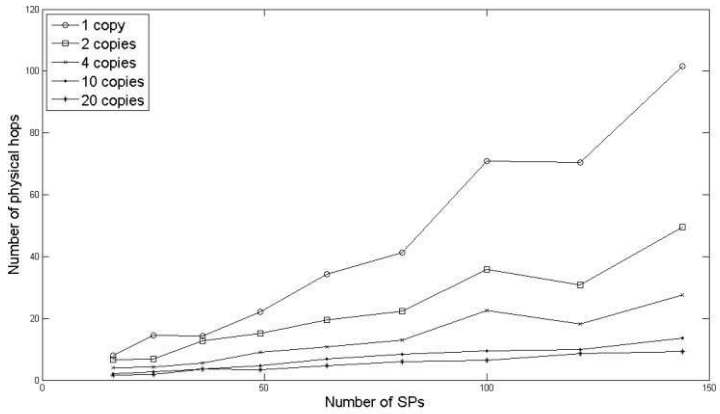

Figure 3-6: Physical hops in Georoy with resource replication. Grid topology.

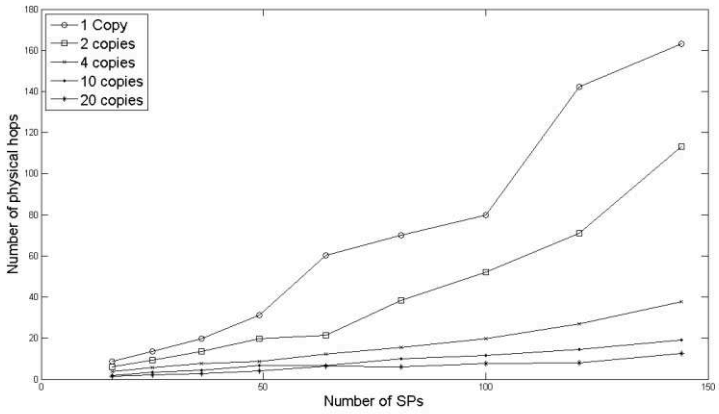

Figure 3-7: Physical hops in Georoy with resource replication. Random topology.

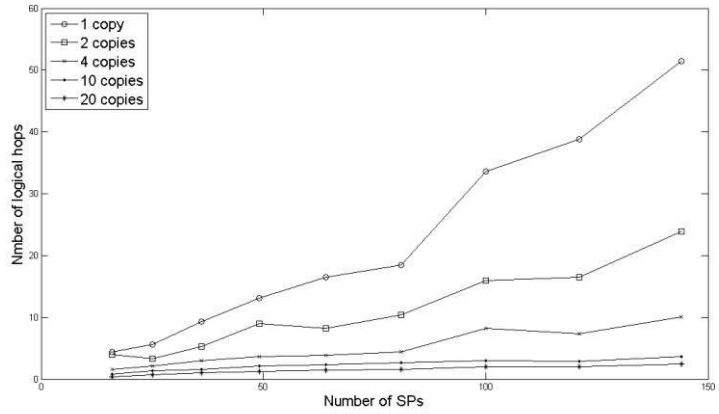

Figure 3-8: Logical hops in Georoy with resource replication. Grid topology.

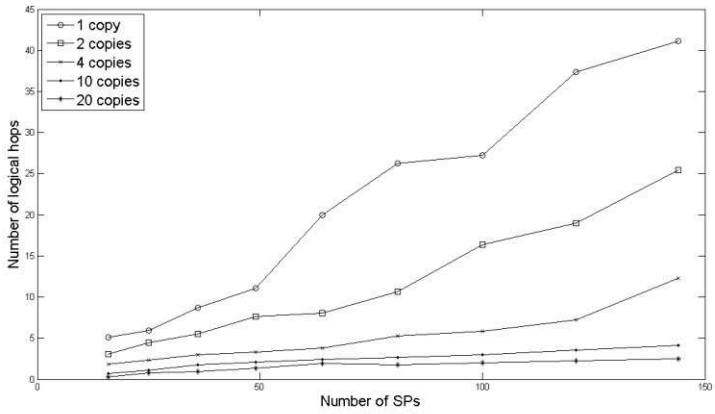

Figure 3-9: Logical hops in Georoy with resource replication. Random topology.

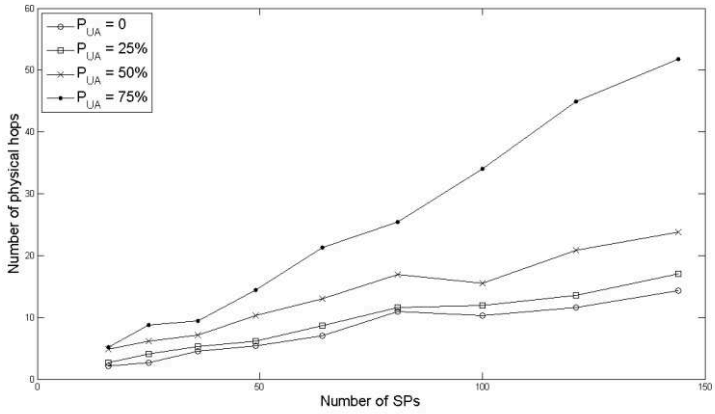

Figure 3-10: Effect of the blacklist on the number of physical hops for different percentage of unavailability. Grid topology.

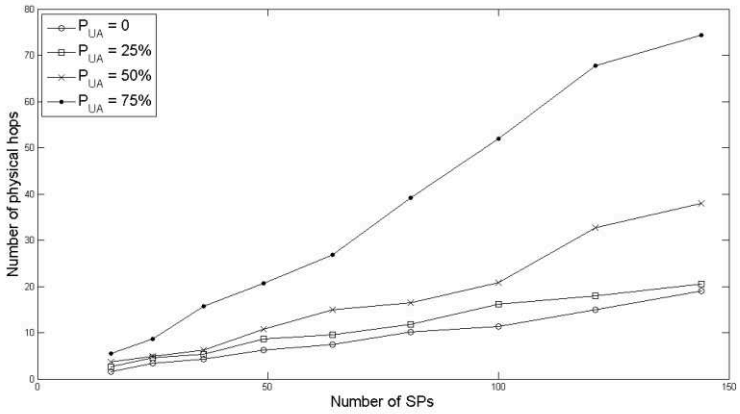

Figure 3-11: Effect of the blacklist on the number of physical hops for different percentage of unavailability. Random topology.

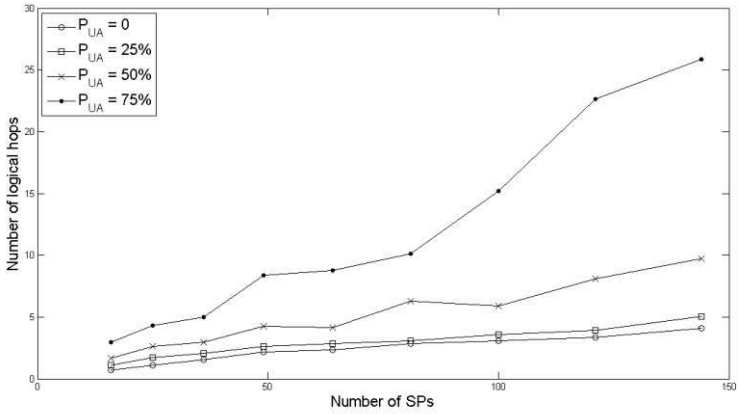

Figure 3-12: Effect of the blacklist on the number of logical hops for different percentage of unavailability. Grid topology.

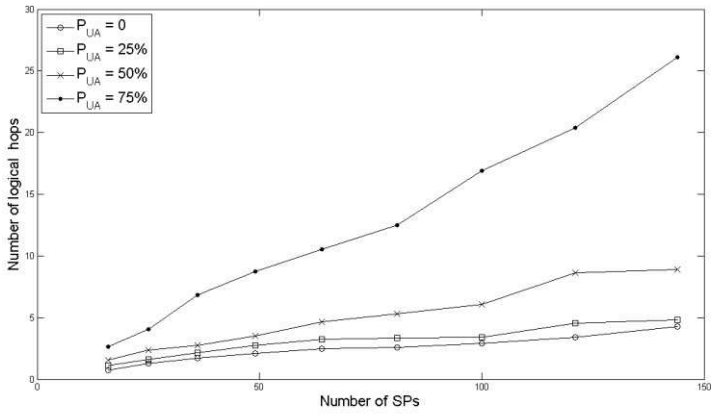

Figure 3-13: Effect of the blacklist on the number of logical hops for different percentage of unavailability. Random topology.

In Figure 3-10-13 we estimated the impact of the use of the blacklist to punish the misbehaving nodes, both in the grid and the random topologies, when 10 copies for each resource are used. More specifically we observed that the impact of misbehaving super peers becomes serious when their percentage increases over 50% since the majority of the copies becomes useless for download, thus increasing the number of logical and physical hops compared to the ideal case.

However, on the other hand, this allows to make zero the percentage of unavailability. This is because the search process will always concern well behaving nodes.

3.5 Conclusions and future works

In this chapter we presented a detailed overview of the Georoy algorithm. As anticipated in the previous chapter, Georoy represents a suitable architectural solution to extend the P2P paradigm to wireless mesh networking. Furthermore, we studied the impact of resource replication and trustworthiness maintenance on performances. The results show that an efficient replication strategy can improve drastically the efficiency of lookup and information retrieval procedures respect with the basic algorithm implementation. An optimal resource dissemination strategy coupled with mechanisms to maintain up-to-date the different copies and procedures able to guarantee good performances in case of nodes churning represent a challenging task to be address in the future works.

4 Opportunistic P2P communications in rural scenarios

In this chapter we consider a rural communication scenario where users move freely within a disconnected rural area. We assume a static infostation deployment is available, for example, using wirelessly connected Internet kiosks, allowing users to connect to the Internet while being located in their closest proximity. Users far from the infostations cannot connect to the Internet unless a data mule comes close to them and a multihop communication can be set up towards the closest infostation. We assume that infostations are connected with each other forming a wireless mesh network. Figure 4-1 shows this reference scenario. In such an environment, rural communications can be allowed during the limited time the isolated user comes into proximity of a data mule which can both perform as a simple relay towards the connected backhaul, or store the requests on the isolated node's behalf and process them while moving. Resource requests performed by remote users can be:

i. issued and retrieved at any time while the user is close to the infostation. In this case, resource search and retrieval are not significantly constrained;
ii. issued and retrieved by an isolated remote node during the limited time the data mule comes into its proximity when a multi hop communication can be set up towards the infostation. This could lead to two different situations. In the first case, when the resource search and retrieval is fast, the resource can be searched for and downloaded during the limited proximity time. In the second case, if the search is not fast enough, the resource will be retrieved next time the data mule comes back. In this case, a pure delay tolerant paradigm is employed and only reliability constraints are met.

In order to locate resources distributed within the network, the various P2P schemes above mentioned can be taken into consideration. However, in delay-tolerant application scenarios, opportunistic inter-contact intervals between mobile and remote users should be exploited at maximum since they represent communication chances. To improve the performance of the network, as proposed already by previous literature in the field, resources available can be replicated so that multiple copies of the same file are distributed by exploiting the mobile users' movements and the opportunistic intercontacts with the infostations or kiosks. We present a performance evaluation and comparison

study of two P2P resource management approaches in the opportunistic scenario described before. We identify a tradeoff between search-retrieval efficiency and algorithm complexity. The impact of using these P2P approaches in such scenario is estimated through NS-2 simulations. The main contributions of this work are related to testing of the performance of two efficient P2P approaches conceived for wireless mesh networks and appropriately extended to cope with the constraints of a DTN scenario.

4.1 Scenario Overview

We address an opportunistic scenario where resources are disseminated across the network and nodes can access them. In the illustration of the scenario, we refer to what is shown in Figure 4-1 where infostations are deployed statically, which allow to set up the resource search. Infostations are connected typically using some wireless links and connections among infostations are considered stable. As an example, a mesh network can provide backhaul connectivity between the Infostations, where every mesh router has the functionality to setup the resource search. Also, there is a certain number of peripheral nodes which can provide and/or search for resources.

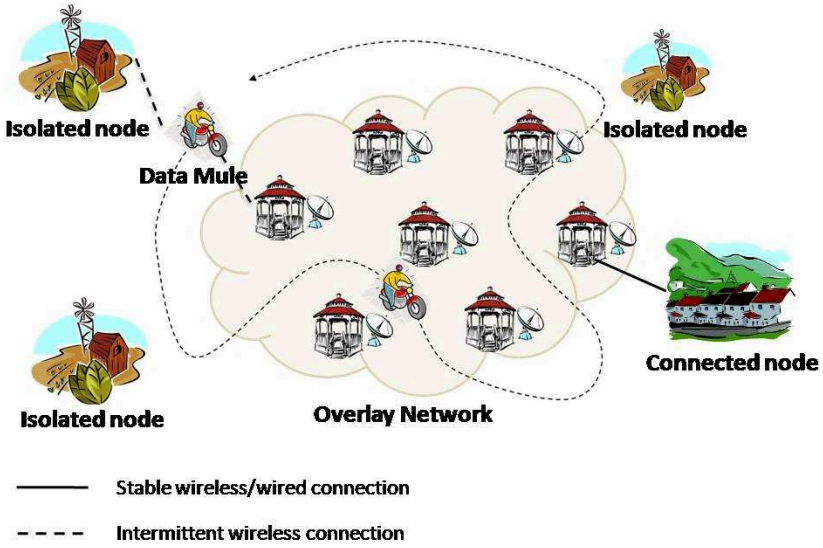

Figure 4-1: Scenario overview.

Some of these peripheral nodes can be isolated and not in the range of any infostation so that their resources cannot be shared and their requests cannot be served directly. We assume that one or more data mules can move around and serve the isolated nodes once they come in their closest proximity. Obviously, the mobile node remains in the proximity of the isolated node for a limited time interval during which resource search must be performed and the resource should be provided to the requesting node. If these two processes are not successfully completed during the limited proximity time, the isolated node cannot exchange data with the rest of the network. A solution to this problem could exploit a delay-tolerant paradigm. In fact, the mobile node can cache the lookup request as issued from the isolated node and keep on

performing the lookup during its tour throughout the network. Once the lookup request is answered successfully, the data mule retrieves the resource and stores it until it comes again in proximity of the isolated node which can then be served. In order to implement the above-presented scenario, we have chosen to compare the performance of Georoy and Bamboo P2P protocols for wireless networks, appropriately extended to cope with the opportunistic networking scenario by including a replication strategy for managing multiple copies of the same source. Speeding up the lookup process is important as the data mule is in close proximity of a given infostation only for a limited time period. This time interval during which the lookup request/response needs to be completed depends on the speed of the data mule and the mobility pattern.

4.2 Resource replication strategy

In P2P networks, the lookup procedure can take very long time when the size of the network increases. This is especially the case when deployed over multi hop wireless networks as for each physical hop, the lookup message needs to contend again for the medium. Therefore, reducing the total number of physical hops traveled directly impacts on the achievable performance. Also, when only a single copy of the resource is available in the network (for worth of simplicity, in the following we will assume that a LP

node provides only a single resource but generalization to the case of multiple resources provided by a node is straightforward.) the provider node could become congested if multiple peers request the resource. Moreover, if the responsible node crashes, the resource will be no longer available. Accordingly, replication of resources can be beneficial since it allows to balance the network traffic among the different replicas' providers. This can reduce the delivery delay both in case of resource lookup and resource delivery. In fact, when more copies of a resource are available in the network, it is expected that the resource can be located in the closer proximity of the requesting node. While a replication mechanism for Bamboo has already been specified, in this paper, we develop a replication strategy for Georoy (only outlined in the previous chapter) which we will describe in the following.

4.2.1 Resource replication in Georoy

In Georoy when a LP storing a resource and located closer to an infostation node, denoted as SP, moves it can decide to replicate its resources with a given probability, P_R, at its old SP. For example, if the LP denoted as *D*, previously located closer to the SP denoted as *B* moves, it can decide if leaving a copy of its resource in *B*'s area or not according to a given probability P_R. Then, when *D* moves and comes into proximity of a new SP called *C*, its resource becomes again available. So the number of copies of

each *D*'s resource into the network are given by $(1 + N_{SPv} \cdot P_R)$ where N_{SPv} represents the number of different SP nodes visited during *D*'s tour in the interest area. In fact, if a node visits many time the same SP, it does not try to replicate its resource at the same node everytime but just once. Replication of a resource requires an update at the Home SP managing the range of keys to which the resource belongs. When the LP node *D* moves and goes out of the coverage area of its responsible SP, if replication happened, *B* will ask one of the other LPs in its coverage area to store the copy of the resource. This will be done through a *put_resource* message. Then *B* will contact through a lookup the corresponding Home SP storing the range of keys the resource belongs to and notifies the availability of a replica of that resource at its site. When the node *D* comes into the proximity of another SP *C*, it will notify its catalog of resources and the SP *C* will keep the Home SP updated through a lookup operation. When a lookup for that resource will be generated, it will be forwarded throughout the Georoy overlay as usual. Two cases can happen.

1) Case 1: the resource is available at one of the SPs traversed along the path going to the Home SP responsible for that resource. In this case, the lookup is positively answered before reaching the Home SP and the resource is located fastly.
2) Case 2: the lookup is forwarded till the Home SP is met but the resource is not located before reaching the Home

SP. In this case, the Home SP owns a list of the SP nodes that have the resource in their catalog. Accordingly, based on the ID of the node who issued the lookup, the Home SP answers with the ID of the SP among those which store the resource that is closer to the ID of the requester. This is because closer IDs in the logical space mean also closer physical location due to the intrinsic property of the Georoy mapping.

Observe that replication implies an increase in the rate of availability of a resource in the network but could cause an increase also in the overhead at network nodes. Accordingly, a mechanism to control the number of replicas of a given resource available in the network should be considered. To this purpose, in Georoy we assume that the oldest copies of a resource are deleted after a time out so that a maximum number of replicas for a resource R_M can be found into the network. To implement this control, Georoy has been modified in the following way. When the Home SP of a resource, which is aware of the number of copies of a resource available in the network and the time they were generated, sees that R_M copies are currently available into the network, as soon as it receives another notification for a new copy of the resource, will accept it and contact the responsible SP for the oldest copy to ask for deleting the resource from the catalog. To this purpose a *delete_resource* message will be sent. The Responsible SP, upon receiving such a notification, contacts the LP storing the copy and,

if it is still in its coverage area, asks for deleting the resource. Accordingly, a *delete_replica* message will be exploited. If the LP moved, the resource is considered no longer available in any case.

To be sure that the available replicas of the resource are still valid, each responsible SP periodically interrogates the LP that is supposed to store the copy of the resource using a beaconing-like approach. If the LP moved without notification, the status of the copy is updated as *parked* at the Home SP and managed as specified in the following section. Accordingly the number of copies of a resource in the network is kept updated.

4.2.2 Resource replication in Bamboo

In Bamboo, a replication mechanism is already incorporated. This is quite simple with respect to Georoy and provides incremental scalability. Basically, a node holding a given resource also caches it within some of its leafset neighbors. This is done according to a number of desired replicas. To this purpose, *put* messages are generated by the node to selected peers among its successors and predecessors. For example, if the desired number of replicas is set to 4, the node generates 4 Bamboo *put* messages destined to 2 random successors and 2 predecessors, achieving a total of 5 resource copies in the network. Therefore, the amount of overhead in the network increases with the number of replicas. It is also important to note that the maximum amount of replicas is

given by the total number of nodes in the leafset (i.e., number of successors and predecessors). This means that in the default scenario where the number of leafsets is configured to 7, a maximum of 15 copies of the resource will be available in the network.

When an existing node leaves the system, it takes the data it has stored with it. Therefore, the redundancy given by the replication strategy guarantees that the resource will be still available in the remaining leafset neighbors. In order to keep the distributed storage consistent, data storage updates are also applied by Bamboo, where a node periodically picks a random node in its leafset and synchronizes the stored data with it. The correspondent node calculates the set among its stored data that should be stored at the peer node, sending this data to it, including the hash values of the data. For certain applications, the number of desired replicas can cause large demands for storage space. This can turn into serious scalability problems when disseminating these replicas to many nodes in the leafset.

4.3 Performance results

In this section we compare the performance of the two protocols, Bamboo and Georoy, in different conditions. We want to better understand their behavior by means of a comparison using two significant scenarios representing the static backhaul of

wireless nodes: grid and random scenarios. Here, a number of SP nodes (i.e., Infostations) are placed within an area of a certain size. The Infostations are static and do not move during the simulations. We vary the number of such stations between 25 and 225. Ns2 v2.26 [37] simulations were run considering a transmission range of 200m, a carrier sense range of 250 m, an area which size R depends on the number of SP nodes as $R = N_{SP} \cdot 10^4$ m^2 and a distance between two SPs in the grid topology equal to 100m. Routing between the connected Infostations uses AODV-UU [38] but different choices are possible. In the random topology, nodes are thrown randomly in the area. We consider infinite buffer space on the replication nodes. We make such choice because if the buffer size is limited, achievable performance may largely depend on buffer replacement strategies, which is a problem outside the scope of this paper. In the random topology case, for each scenario identified by the number of nodes, we tested 5 different random topologies and for each topology we performed 100 random lookups. Average values and confidence intervals (when applicable) were reported for the following performance metrics being investigated:

a) number of **logical hops** traveled in the overlay network to perform a lookup for a specific resource;
b) corresponding number of **physical hops** traveled in the physical network to perform a lookup for a specific resource as a consequence of the logical path followed;

c) **lookup delay** needed for the lookup to reach the node who stores information about the requested resource (we only consider here correctly completed lookups);
d) **percentage of lookups correctly completed**;
e) **stretch factor**, that is, the ratio between the number of physical hops needed to complete the lookup as a consequence of the logical hops traversed and the number of physical hops going end-to-end according to a shortest path approach.

In the first part of the evaluation, we focus on the impact of the network size on the scalability of the lookup procedure. We then evaluate the impact of the replication technique. Finally, we evaluate the impact of the use of a data mule on the achievable performance in terms of resources download.

4.3.1 Impact of network size

In Figure 4-2 we show the number of logical hops traveled when employing the two algorithms. By comparing the results we observe that, in general, Bamboo results in a smaller number of logical hops as compared to Georoy. This is related to the fact that the amount of overlay routing information used by Bamboo (i.e., leafset and routing table) is higher if compared to Georoy which limits the number of existing logical links to 7.

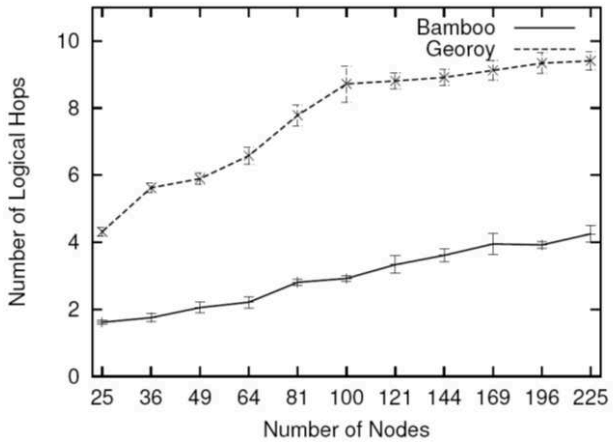

Figure 4-2: Comparison between the number of logical hops in Georoy and Bamboo in grid topology.

Therefore, Bamboo can more easily identify a requested resource as it has more routing information available. In contrast, the number of physical hops mainly impacts on the lookup performance. This is because this parameter determines the number

of forwarding operations a packet needs to undergo in the wireless multi hop network to reach the destination (i.e., the node holding the resource).

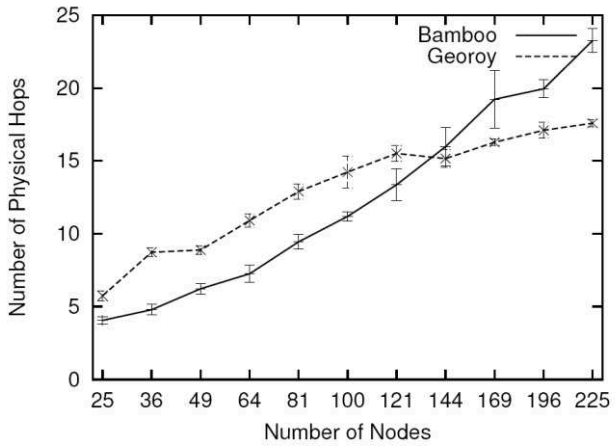

Figure 4-3: Comparison between the number of physical hops in Georoy and Bamboo in a grid topology.

As the network size grows, also the number of physical hops needed to complete a lookup increases (see Figure 4-3). However, an interesting observation is that for larger topologies, the number of physical hops is in general lower when using Georoy as compared to Bamboo. This is because, due to the overlay addressing scheme in Georoy, the logical and physical topologies are tightly coupled so that the logical path does not differ much from the physical one.

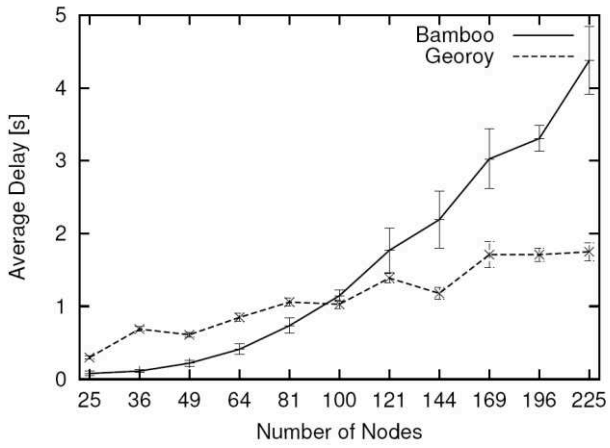

Figure 4-4: Comparison between the delay in Georoy and Bamboo in a grid topology.

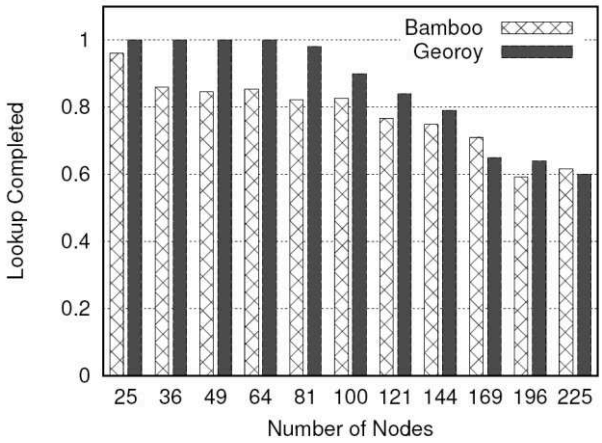

Figure 4-5: Comparison between the percentage of lookups completed in Georoy and Bamboo in a grid topology.

In fact, for large network topologies, the ratio between the physical and logical hops is around 2 for Georoy and rises to 5 for Bamboo. Since the formation of the overlay network is independent of the physical location of the nodes in Bamboo, for larger topologies the probability that a peer selects a close logical neighbor located far away in the physical topology is higher. This results in longer routes when topologies are larger. Also, note that the variance for the physical hops is much smaller in Georoy compared to Bamboo. This is again due to the addressing scheme of Bamboo, which randomly selects nodes in the overlay as neighbors, although they might be actually far away in terms of physical distance. In multi hop wireless networks, the more hops a packet is forwarded, the larger the delay and, in general, the higher the packet loss probability. This is because at every intermediate node, the packet needs to compete for medium access and collisions due to, for example, hidden nodes might lead to frequent retransmissions and consequently high packet loss. The impact of an increase in the number of physical hops traveled in case of large topologies can be seen in the average lookup delay comparison shown in Figure 4-4. Here, we can see that for smaller topologies, Bamboo outperforms Georoy as less physical hops are required. However, due to the efficiency of its addressing scheme, the increase in the number of physical hops is smaller for larger topologies in Georoy, compared to Bamboo. Therefore, Georoy provides better lookup delays with larger topologies. Interestingly, Georoy shows smaller number of

physical hops as compared to Bamboo when network size is larger than 144 nodes. However, the lookup delay of Bamboo is smaller as compared to Georoy already at a network size of about 100 nodes. This apparent discrepancy can be explained due to the fact that the random distribution of requests can turn into a different load on the links. There might be situations where the number of physical hops is a bit smaller for one protocol, but the load on the links might be different resulting in an advantage for the other protocol in terms of delay. Another interesting observation is that the number of successfully completed lookups decreases as network size increases (see Figure 4-5). By increasing the number of nodes in the network we also increase the amount of messages exchanged (management traffic required to maintain the overlay plus key lookup request/replies) among the nodes and consequently the wireless contention for the medium. Also, when lookup packets traverse more hops, they need to compete more often for medium access and the probability to collide due to, for example, hidden nodes is higher. Interestingly, the number of completed lookup requests is smaller for Bamboo as compared to Georoy, even for small topologies. This can be attributed to the fact that the management traffic of Bamboo is significantly higher. Such high-management traffic leads to more load and contention leading to higher chance that the lookup request cannot be completed correctly. In Bamboo, in this case the lookup request is retransmitted a limited amount of time until the agent gives up and

declares the request as not successful. The stretch factor presented in Figure 4-6 shows that both protocols can satisfy lookup requests with a limited increase in the number of hops traversed when compared to the shortest path approach. As we have seen in Figure 4-3, Georoy needs fewer hops to forward a lookup request to the destination when the network is composed of 144 nodes or more. Consequently, the stretch factor of Georoy is smaller compared to Bamboo at large network sizes. When considering the random topologies, similar conclusions can be drawn. However observe that, in the random case, nodes are not distributed on the vertices of a grid, so physical proximity can help to reduce the number of physical hops and, thus, decrease delay significantly as evident in Figure 4-8 and Figure 4-9. In fact when performing a lookup operation, one can move in any direction to a neighbor node which is not constrained to be located on a grid vertex. In addition, due to the random nature of the node location, we could observe more clustering of nodes as compared to a grid scenario. Therefore, as nodes are more close to each other in most of the area, less physical hops are required, thus implying less delay to complete the lookup operation. Clearly, due to the randomness in node location, there is more variability in the number of physical hops and delay. The logical hops instead do not vary much as compared to the grid scenario (see Figure 4-7).

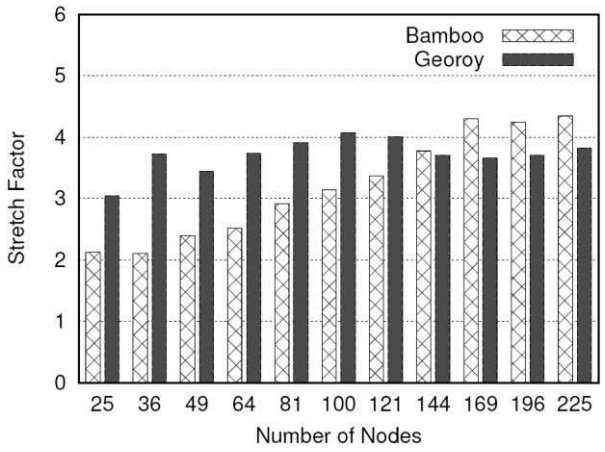

Figure 4-6: Comparison between the stretch factor in Georoy and Bamboo in a grid topology.

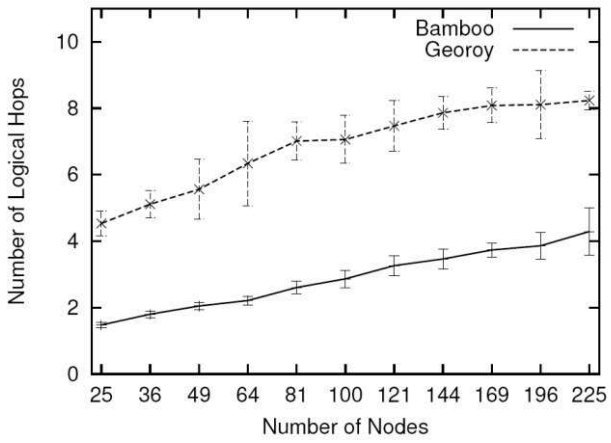

Figure 4-7: Comparison between the number of logical hops in Georoy and Bamboo for random topology.

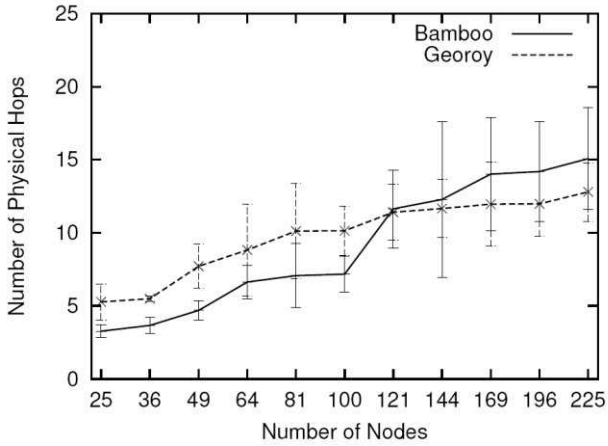

Figure 4-8: Comparison between the number of physical hops in Georoy and Bamboo for random topology.

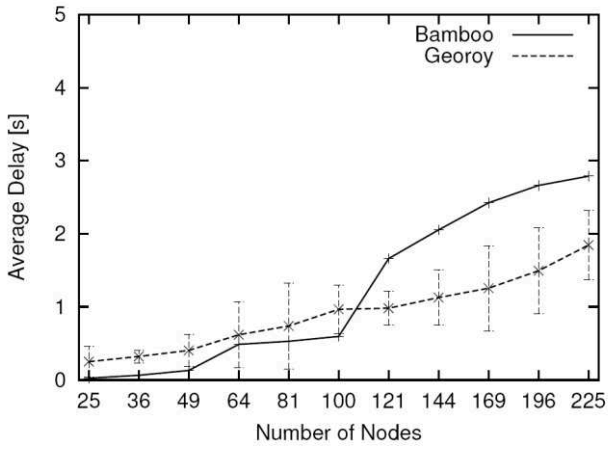

Figure 4-9: Comparison between the delay in Georoy and Bamboo for random topologies.

4.3.2 Impact of replication

Besides the impact of network size in grid and random topologies, another important point that we address is to determine the benefit of using a replication mechanisms in opportunistic scenarios. We start by looking at the impact of having different number of replicas as a way to speed up the resource lookup process. In our experiments we considered that both in Georoy and Bamboo each resource was replicated at 3, 5, or 7 different nodes. We assume a random waypoint mobility of the LP node providing the resource and, consequently, the replicas of the resource are randomly distributed in Georoy while replicas are assigned to random nodes in the leafset in Bamboo, independently of the LP movement. In Figure 4-10 and Figure 4-11 we observe that, upon increasing the number of copies of a resource, both the number of logical and physical hops slightly decrease. As expected this is because, when increasing the number of replicas, the probability of finding the resources closer raises as well. As a result, when using more replicas, the delay to complete a resource lookup can be reduced as evident looking at Figure 4-12. Also, consider that in Bamboo no significant variations in the number of logical hops as a consequence of a change in the number of resource replicas are met.

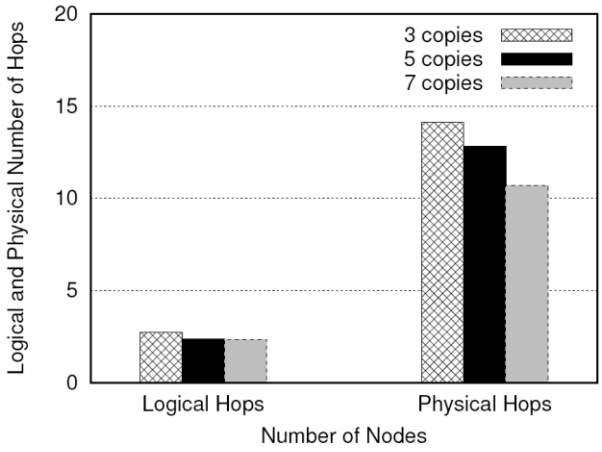

Figure 4-10: Number of logical and physical hops in Bamboo in a grid topology with 225 nodes.

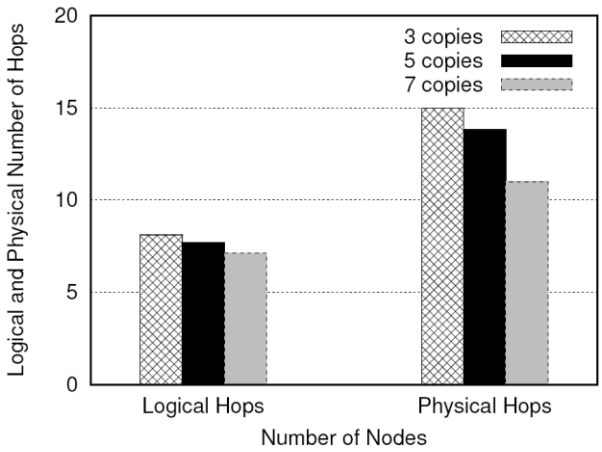

Figure 4-11: Number of logical and physical hops in Georoy in a grid topology with 225 nodes.

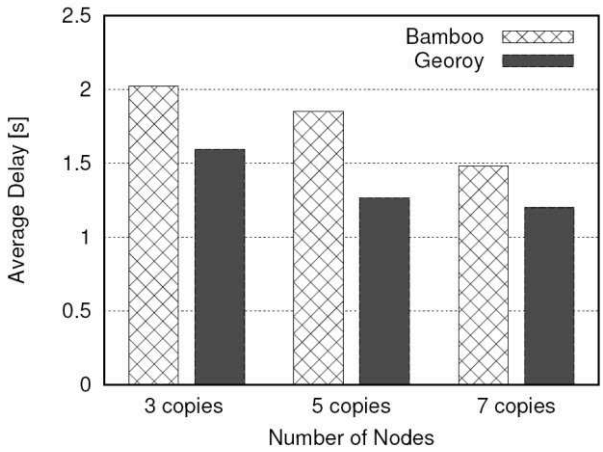

Figure 4-12: Delay in Georoy and Bamboo in a grid topology with 225 nodes.

The reason for this behavior is to be searched in the replication mechanism which in Bamboo disseminates replicas randomly at nodes in the leafset which are thus very close in the logical space but could not give meaningful help in speeding up the lookup procedure. Also observe that in Georoy it is sufficient to use a controlled number of replicas (i.e., higher than or equal to 5) to achieve quite stable performance.

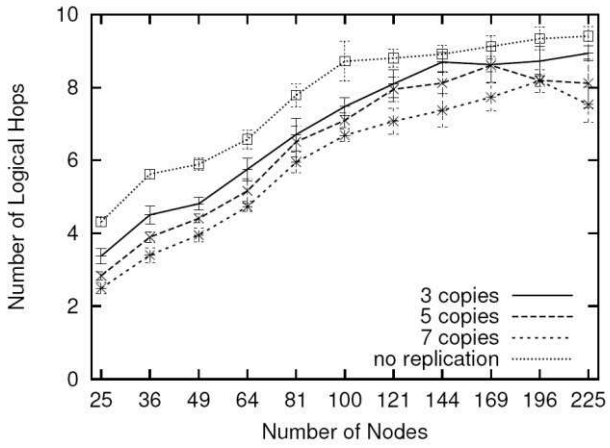

Figure 4-13: Number of logical hops in Georoy in a grid topology exploiting resource replication.

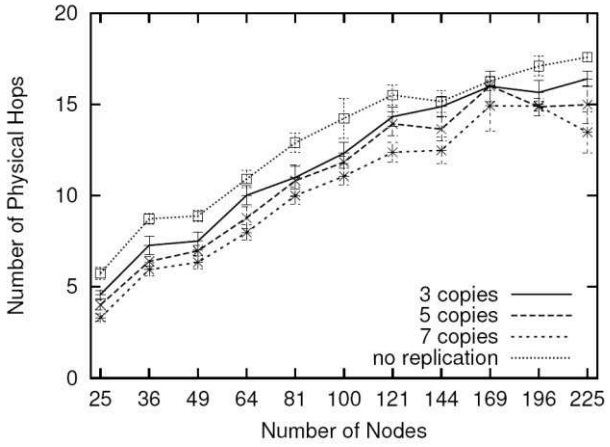

Figure 4-14: Number of physical hops in Georoy in a grid topology exploiting resource replication.

4.3.3 Impact of data mule mobility

Finally, we wanted to test how the two protocols behave in case of a disconnected scenario where an isolated node wants to perform a lookup but can only execute it during the limited time spent by a data mule, who travels around the network area, in its coverage range. In particular in this case we estimated the delay for a resource retrieval. We assumed a mobile data mule moving with a velocity variable in [4 Km/h (pedestrian case), 10Km/h (vehicular case 1) and 25 Km/h (vehicular case 2)]. An isolated node, in the best case, will have the data mule in its coverage area for a time equal to $2 \cdot R/v$ where R is the transmission range and v is the data mule velocity. We assumed a retrieval for a file of size 2MB with links of capacity equal to 1 Mbps. We considered a variable number of retransmissions on each link in [1, 3]. Accordingly in Figure 4-15 and Figure 4-16, we show:

a) the maximum delay taken for performing the lookup and retrieving the file in case of 1 retransmission on each link (delay 1 retr);
b) the maximum delay taken for performing the lookup and retrieving the file in case of 3 retransmissions on each link (delay 3 retr);
c) the maximum available time for lookup and retrieval depending on the data mule velocity (max delay).

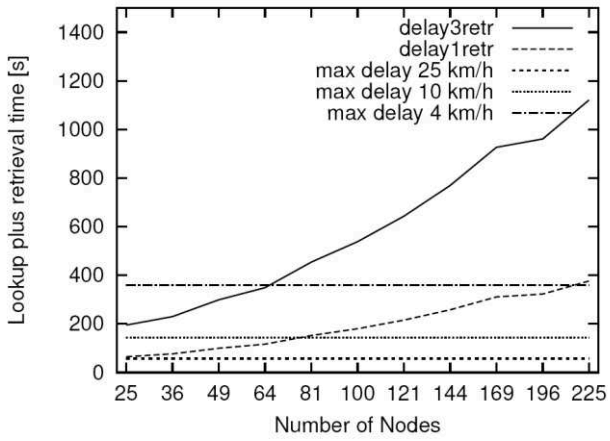

Figure 4-15: Delay-tolerant networking statistics in Bamboo.

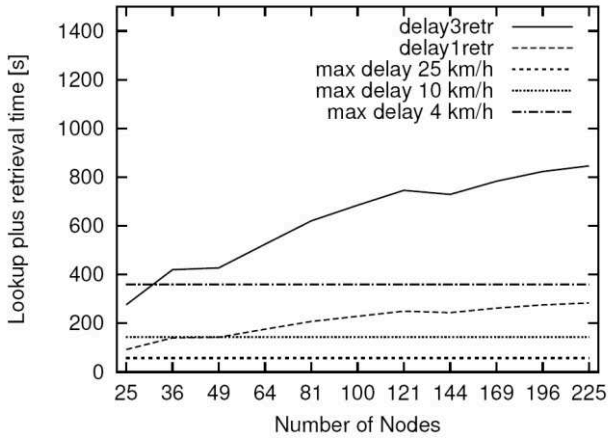

Figure 4-16: Delay-tolerant networking statistics in Georoy.

Comparing the two plots we observe that for both Georoy and Bamboo the resources can be retrieved during the limited proximity time if the data mule moves around 4 Km/h. Instead, when the velocity of the mule is higher (10 or 25Km/h), the percentage of retrieved resources during the contact time decreases and the delivery will be delayed of an amount equal to the intercontact time, that is, the time passed since previous exit until next entry of the mule into the coverage area of the isolated node. Supposing to employ a random way-point model for the data mule movements, the CDF of intercontact time [39] is shown in Figure 4-17 and Figure 4-18 by varying the number of super peers and the mule's velocity, respectively. Looking at the curves related to the velocity of mule around 10Km/h, we observe that Georoy can complete the retrieval of a resource during the proximity time when the number of SPs is lower than or equal to 49; in Bamboo, instead, the delivery can be satisfied when the number of SPs is lower than 81.

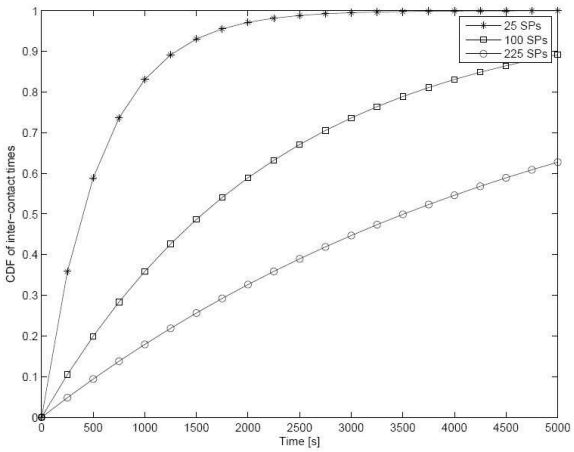

Figure 4-17: CDF of intercontact time for different number of SPs.

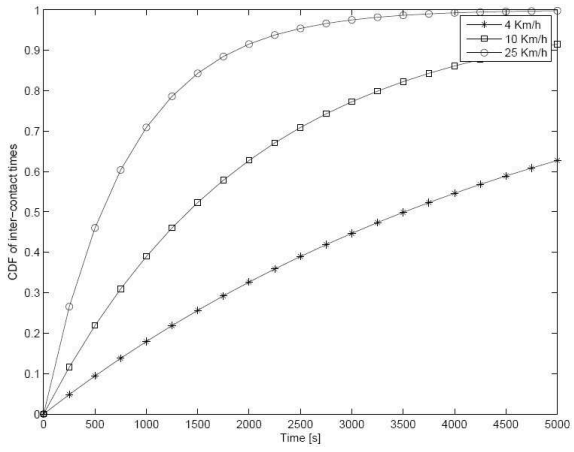

Figure 4-18: CDF of intercontact time for different data mule's velocity.

Finally, in Figure 4-19 we show the percentage of downloads completed by an isolated node during the transit period of the data mule, when the latter moves at 10Km/h, by varying the number of copies for each resource. As we observed, upon increasing the number of replicas, the retrieval procedure can be speed up: without replication, Georoy is able to efficiently exploit the limited proximity time for exchanging data until a maximum number of SPs equal to 49; exploiting the random dissemination of the resources, instead, this threshold can be increased until 81 SPs employing 7 copies for each resource.

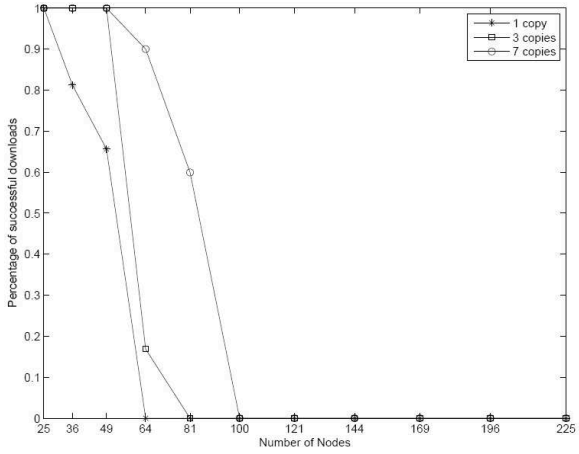

Figure 4-19: Replication effectiveness in resource downloading for Georoy.

Conclusions of this analysis are the following:

i. Bamboo performs better than Georoy in small to medium size topologies both grid or random. This is due to the more complete view of the overlay given by the larger overlay routing information, which also requires higher management traffic. When network size increases, Georoy overcomes Bamboo in performance due to the location aware addressing scheme.
ii. Random topologies lead to a reduction in the number of hops and, thus, in the delay with respect to more regular cases like grid topology. This is mainly due to the clustering of nodes, which reduces the number of required physical hops.
iii. Bamboo in general exhibits a lower number of lookups completed successfully due to its high overhead.
iv. In opportunistic scenarios, where a data mule travels around and helps to connect remote nodes to infostations, when the data mule does not move too fast both protocols can allow lookup and delivery during the limited proximity time although Bamboo is more convenient also for slightly higher velocities. Performance improves when the download volume reduces or the data mule moves slower.

4.4 Conclusions

In this chapter we addressed the problem of efficient content distribution and resource retrieval in opportunistic challenged scenarios. The latter are characterized by intermittent connectivity and, thus, use of traditional P2P approaches proposed for reliable and connected wireless networks does not always show effectiveness in these networks. Accordingly, we considered two efficient P2P schemes for wireless networks and enhanced them by introducing procedures to allow increasing scalability and reliability by use of multiple replicas of the same resource in the network and management of network disconnections. Performance results were aimed at comparing the performance of the two algorithms (Bamboo and Georoy) in both the case of static connected networks and delay-tolerant scenarios.

Our proposed extension focuses on scenarios where we have a set of infostations (SPs) which are connected through, for example, some backhaul wireless mesh network. In this case, the proposed techniques such as the replication strategy in Georoy together with the data mule concept, allow to improve performance with respect to the case of lack of replication. However, for a fully delay-tolerant networking scenario where no infrastructure is available and all nodes move around freely, the backhaul would be no longer connected all the time. Depending on the amount of connectivity, one can then question if such structured P2P approach would still

be feasible for a fully disconnected DTN which would rather require physical contacts between SPs in order to exchange information. We argue that when only a few SPs are mobile, the structured P2P approach would still be feasible due to the redundancy of the wireless mesh backhaul, given enough replication is in place. When more and more SPs roam around leading to temporarily sparse deployments, the overlay structure will, at some point, no longer be maintainable and the protocols will not be able to cope with the harsh environment. In such case, epidemic information dissemination resorting to some form of broadcasting could lead to a better performance. However, at what point of mobility/sparse deployment structured P2P approaches fail to deliver suitable performance is out of the scope of the work and should be related also to the specific application scenario being considered.

5 Conclusions

Peer-to-Peer architectures represent an interesting and promiscuous research area in the field of wireless mesh networks and they will play a key role in the diffusion of the mesh paradigm meeting the people's growing need of communication and resource sharing anywhere and anytime.

In this dissertation we briefly presented in Chapter 1 the emerging standards for wireless mesh networks at personal, local and metropolitan scale; then we introduced the opportunistic networking as emerging paradigm in the information and communication technology. In Chapter 2 we described the most popular P2P overlay schemes proposed in literature highlighting the most important features of each of them. Proposed to address the requirements of large scale wired networks, they do not represent efficient solutions for wireless environments. Nevertheless, these schemes represent an irreplaceable source of ideas in designing effective architectures allowing the extension of the P2P paradigm to wireless mesh networks. With this aim we drew the conclusion that a two-tier hierarchical, decentralized, and DHT-based structure can represent a suitable solution.

Under this perspective, Chapter 3 focused on a location-aware variant of the Viceroy protocol, called Georoy, matching the above mentioned features. Our contribution consisted in studying the impact of resource replication and trust preservation on lookup performances in grid and random topologies. In particular, we showed that the introduction of resource replicas can reduce drastically the lookup procedure latency.

In Chapter 4 we addressed the problem of efficient content distribution and resource retrieval in opportunistic challenged environments such as rural communications, comparing the performances of Georoy and Bamboo algorithms in both the case of static connected networks and delay-tolerant scenarios.

Thanks to this study we argued that when only a few SPs (or mesh peers) are mobile or intermittent, the structured P2P approach would still be feasible exploiting opportune improvements such as resource replication strategy, location awareness, churning management, and so on; instead, when more and more mesh peers roam around or show intermittent working leading to temporarily sparse deployments, the structured P2P overlay will fail to deliver suitable performance.

References

[1]. L. Galluccio, S. Palazzo, C. Rametta. "**On the efficiency and trustworthiness of DHT-based P2P search algorithms in mobile wireless networks**". In Proceedings of International Conference on Ultra Modern Telecommunications and Workshops, Saint Petersburg, 2009.

[2]. M. C. Castro, L. Galluccio, A. Kassler, S. Palazzo, C. Rametta. "**On the comparison between performance of DHT-based protocols for opportunistic networks**". In Proceedings of Future Network and Mobile Summit, Florence, Italy, 2010.

[3]. M. C. Castro, L. Galluccio, A. Kassler, C. Rametta. "**Opportunistic P2P communications in delay-tolerant rural scenarios**". EURASIP Journal on Wireless Communications and Networking, 2011.

[4]. E. K. Lua , J. Crowcroft , M. Pias , R. Sharma , S. Lim. "**A Survey and Comparison of Peer-to-Peer Overlay Network Schemes**". IEEE Communications Surveys and Tutorials, March 2004.

[5]. E. Meshkova, J. Riihijarvi, M. Petrova, P. Mahonen. "**A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks**". Elsevier Computer Networks, August 2008.

[6]. L. Galluccio, G. Morabito, S. Palazzo, M. Pellegrini, M. E. Renda, and P. Santi. "**Georoy: a location-aware enhancement to Viceroy peer-to-peer algorithm**". Computer Networks, vol. 51, no. 8, 2007.

[7]. I. F. Akyildiz, X. Wang, and W. Wang. "**Wireless mesh networks: a survey**". Computer Networks, vol. 47, no. 4, 2005.

[8]. R. Bruno, M. Conti, and E. Gregori. "**Mesh networks: commodity multihop ad hoc networks**". IEEE Communications Magazine, March 2005.

[9]. M. J. Lee, J. Zheng, Y. Ko, and D. M. Shrestha. "**Emerging standards for wireless mesh technology**". IEEE Wireless Communications, April 2006.

[10]. I. F. Akyildiz, X. Wang. "**Wireless mesh networks**". John Wiley & Sons, 2009.

[11]. S. M. Faccin, C. Wijting, J. Kneckt, A. Damle. "**Mesh WLAN networks: concept and system design**". IEEE Wireless Communications, April 2006.

[12]. L. Pelusi, A. Passarella, and M. Conti. "**Beyond MANETs: dissertation on Opportunistic Networking**". IIT-CNR Technical Report, May 2006.

[13]. P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. "**Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet**". In Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X '02), October 2002.

[14]. P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. "**Pocket switched networks and human mobility in conference environments**". In Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networking and Related Topics (WDTN '05), Philadelphia, Pa, USA, August 2005.

[15]. **Reality Mining project**, http://reality.media.mit.edu.

[16]. **Haggle project**, ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/fire/projects-haggle_en.pdf.

[17]. T. Spyropoulos, K. Psounis, and C. S. Raghavendra. "**Single copy routing in intermittently connected mobile networks**". In Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04), October 2004.

[18]. M. Grossglauser and D. N. C. Tse. "**Mobility increases the capacity of ad hoc wireless networks**". IEEE/ACM Transactions on Networking, vol. 10, no. 4, 2002.

[19]. J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang. "**Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks**". In Proceedings of the IEEE 61st Vehicular Technology Conference (VTC '05), June 2005.

[20]. J. Leguay, T. Friedman, and V. Conan. "**DTN routing in a mobility pattern space**". In Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networking and Related Topics (WDTN '05), Philadelphia, Pa, USA, August 2005.

[21]. M. Musolesi, S. Hailes, and C. Mascolo. "**Adaptive routing for intermittently connected mobile ad hoc networks**". In Proceedings of the Sixth IEEE International Symposium on a

World of WirelessMobile andMultimedia Networks (WoWMoM'05), Los Alamitos, Calif, USA, June 2005.

[22]. J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. "**Max-Prop: routing for vehicle-based disruption-tolerant networks**". In Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06), Barcelona, Spain, April 2006.

[23]. K. Scott and S. Burleigh. "**Bundle protocol specification**". Tech. Rep., NASA Jet Propulsion Laboratory, November 2007.

[24]. J. Seguì and E. Jennings. "**Delay tolerant networking-bundle protocol simulation**". In Proceedings of the 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT '06), July 2006.

[25]. Y. Gong, Y. Xiong, Q. Zhang, Z. Zhang,W.Wang, and Z. Xu. "**Anycast routing in delay tolerant networks**". In Proceedings of the Global Telecommunications Conference (GLOCOM '06), December 2006.

[26]. W. Zhao, M. Ammar, and E. Zegura. "**Multicasting in delay tolerant networks: semantic models and routing algorithms**". In Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networking and Related Topics (WDTN '05), Philadelphia, Pa, USA, August 2005.

[27]. Q. Ye, L. Cheng, M. C. Chuah, and B. D. Davison. "**OSmulticast: on-demand situation-aware multicasting in disruption tolerant networks**". In Proceedings of the IEEE 63rd Vehicular Technology Conference (VTC '06), Melbourne, Australia, July 2006.

[28]. K. Scott. "**Disruption tolerant networking proxies for onthe-move tactical networks**". In Proceedings of the Military Communications Conference (MILCOM '05), October 2005.

[29]. A. Balasubramanian, Y. Zhou, W. B. Croft, B. N. Levine, and A. Venkataramani. "**Web search from a bus**". In Proceedings of the 2nd ACMWorkshop on Challenged Networks (CHANT '07), September 2007.

[30]. C. Plaxton, R. Rajaraman, and A. Richa. "**Accessing nearby copies of replicated objects in a distributed environment**". In Proceedings of the 9th Annual ACM Symposium on Parallel Algorithms and Architectures, 1997.

[31]. A. Rowstron and P. Druschel. "**Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems**". In Proceedings of the Middleware, 2001.

[32]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "**A scalable content addressable network**". In Processings of the ACM SIGCOMM, 2001, pp. 161–172.

[33]. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. "**Chord: A scalable peer-to-peer lookup protocol for internet applications**". IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 17–32, 2003.

[34]. B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. "**Tapestry: A resilient global-scale overlay for service deployment**". IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, pp. 41–53, January 2004.

[35]. **Napster**. Available: http://www.napster.com/

[36]. Gnutella development forum, **the gnutella v0.6 protocol**. Available at: http://groups.yahoo.com/group/the gdf/files/

[37]. **Gnutella ultrapeers**. Available at: http://rfc-gnutella. sourceforge.net/Proposals/Ultrapeer/Ultrapeers.htm/

[38]. P. Maymounkov and D. Mazieres. "**Kademlia: A peer-to-peer information system based on the xor metric**". In Processings of the IPTPS, Cambridge, MA, USA, February 2002, pp. 53–65.

[39]. D. Malkhi, M. Naor, and D. Ratajczak. "**Viceroy: a scalable and dynamic emulation of the butterfly**". In Processings of the ACM PODC'02, Monterey, CA, USA, July 2002, pp. 183–192.

[40]. D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. "**Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web**". In Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, May 1997, pp. 654–663.

[41]. "**Secure hash standard**". NIST, U.S. Dept. of Commerce, National Technical Information Service FIPS 180-1, April 1995.

[42]. H. J. Siegel. "**Interconnection networks for simd machines**". Computer, vol. 12, no. 6, pp. 57–65, 1979.

[43]. I. Abraham, D. Malkhi, and O. Dubzinski. "**Land: Stretch (1+epsilon) locality aware networks for dhts**". In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA'04), New Orleans, LA., USA, 2004.

[44]. I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. "**Freenet: A distributed anonymous information storage and**

retrieval system. Freenet White Paper.". Available at: http://freenetproject.org/freenet.pdf

[45]. **Fasttrack** peer-to-peer technology company. Available at: http://www.fasttrack.nu/

[46]. **Kazaa** media desktop. Available at: http://www.kazaa.com/

[47]. **Bittorrent**. Available at: http://bitconjurer.org/BitTorrent/

[48]. **The overnet file-sharing network**. Available at: http://www.overnet.com/

[49]. **Overnet/edonkey2000**. Available at: http://www.edonkey2000.com/

[50]. "**Public key cryptography for the financial services industry – part 2: The secure hash algorithm (sha-1)**". American National Standards Institute, Tech. Rep. American National Standard X9.30.2-1997, 1997.

[51]. P. Ganesan, Q.Sun, and H. Garcia-Molina. "**Yappers: A peer-to-peer lookup service over arbitrary topology**". In Proceedings of the IEEE Infocom 2003, San Francisco, USA, March 30 - April 1 2003.

[52]. Q. Lv, S. Ratnasamy, and S. Shenker. "**Can heterogeneity make gnutella scalable ?**". In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS), Cambridge, MA, USA, February 2002.

[53]. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. "**Making gnutella-like p2p systems scalable**". In Proceedings of the ACM SIGCOMM, Karlsruhe, Germany, August 25-29 2003.

[54]. A. Barabasi, R. Albert, H. Jeong, and G. Bianconi. "**Power-law distribution of the world wide web**". Science, vol. 287, 2000.

[55]. R. Albert, H. Jeong, and A. Barabasi. "**Diameter of the world wide web**". Nature, vol. 401, pp. 130–131, 1999.

[56]. **SARI project**. http://edev.media.mit.edu/SARI/

[57]. **COW project**. http://www.vidal.org.in/

[58]. K. Fall. "**A delay-tolerant network architecture for challenged internets**". In Proceedings of SIGCOMM 2003.

[59]. S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. "**Handling churn in a DHT**". In Proceedings of the Annual Technical Conference (USENIX '04), June 2004.

[60]. S. Rhea, B. Godfrey, B. Karp et al. "**OpenDHT: a public DHT service and its uses**". ACM SIGCOMM Computer Communication Review, vol. 35, no. 4, pp. 73–84, 2005.

[61]. M. C. Castro, E. Villanueva, I. Ruiz, S. Sargento, and A. J. Kassler. "**Performance evaluation of structured P2P over wireless multi-hop networks**". In Proceedings of the 2nd International Conference on Sensor Technologies and Applications (SENSORCOMM '08), August 2008.

[62]. **Network Simulator 2**, http://www.isi.edu/nsnam

[63]. **Uppsala implementation of the AODV protocol**, http://sourceforge.net/projects/aodvuu/files/AODV-UU/0.9.5/aodvuu-0.9.5.tar.gz/download

[64]. B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M.Wawrzoniak, and M. Bowman. "**PlanetLab: An Overlay Testbed for Broad-Coverage Services**". In Proceedings of ACM SIGCOMM, Karlsruhe, Germany, 2003.

[65]. MatLab. http://www.mathworks.com

[66]. **IEEE 802.15.5 WPAN Mesh Networks**. Available at: http://ieee802.org/15/pub/05/15-05-0260-00-0005-802-15-5-mesh-networks.pdf

[67]. **IEEE 802.11: Wireless LAN**. Available at: http://ieee802.org/11

[68]. **IEEE 802.16: Wireless MAN**. Available at: http://ieee802.org/16

[69]. H. Schulze, K. Mochalski. "**Internet study 2008/2009**". Available online: http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf