Measurement Campaign on Connectivity of Mesh Networks formed by Mobile Devices

Group No. 07gr1112

Beatrice Pietrarca
Giovanni Sasso

May 2007

Assoc. Prof. Frank Fitzek
PhD Student Gian Paolo Perrucci

Aalborg University

Aalborg University

E-Studyboard

Mobile Communications 10th semester

TITLE:

Measurement Campaign on Connectivity of Mesh Networks formed by Mobile Devices

THEME:

Communication, Cooperation and Wireless Networks

PROJECT PERIOD:

1^{st} February - 31^{st} May 2007

PROJECT GROUP:

07gr1112

GROUP MEMBERS:

Beatrice Pietrarca
Giovanni Sasso

SUPERVISORS:

Frank Fitzek
Gian Paolo Perrucci

Number Of Duplicates: 6

Number Of Pages In Report: 38

Number Of Pages In Appendix: 12

Total Number Of Pages: 50

Abstract

In the last few years, many research works have focused on mobile wireless networks. In particular, wireless mesh networks are attracting significant interest. Mesh networks employ Peer-to-Peer (P2P) communication in the form of multi-hop forwarding among wireless intermediate nodes to relay information from one point to another. In this work, we exploit the connectivity among mobile devices to form mesh networks. The Short Range (SR) technology under investigation is Bluetooth (BT), as nowadays most of mobile devices are equipped with BT. We present a measurement campaign regarding the availability of BT devices and SR links among them. Several scenarios are evaluated, mostly public places like airports, shopping malls, trains, buses. The aim is to extract information about the possible size of the mesh network, the types of constituting nodes, the stability of connection, as well as considerations about the time needed for the network organization.

Contents

List of Figures

List of Tables

List of Abbreviations

AFH Adaptive Frequency Hopping

AP Access Point

BER Bit Error Rate

BT Bluetooth

BT ADDR Bluetooth Device Address

BS Base Station

CDF Cumulative Distribution Function

ICDF Inverse Cumulative Distribution Function

CH Cluster Head

CoD Class of Device/Service

DOM Document Object Model

GPRS General Packet Radio Service

GSM Global System for Mobile Communications

ISDN Integrated Services Digital Network

LAN Local Area Network

LOS Line Of Sight

MAC Media Access Control

MaDCl Major Device Class

MaSCl Major Service Class

MiDCl Minor Device Class

MS Mobile Station

PAN Personal Area Network

PC Personal Computer

PDA Personal Digital Assistant

PDF Probability Density Function

P2P Peer-to-Peer

PSTN Public Switched Telephone Network

QoS Quality of Service

TDMA Time Division Multiple Access

SET Similar-Enhanced Transfer

SR Short Range

TDD Time Division Duplex

UMTS Universal Mobile Telecommunication System

WLAN Wireless LAN

XML Extensible Markup Language

Preface

This report is the result of the 10^{th} semester project work carried out between the 1^{st} February and the 31^{st} May 2007 by the students of the Mobile Communications group 07gr1112 at the Department of Communication Technology, Aalborg University, Denmark.

This project report presents a detail study on the connectivity level of Bluetooth (BT) devices in mobile mesh networks. Measurements are obtained by using a BT mobile device equipped with Python S60.

Initially, a theoretical analysis of the core concepts including the basics of P2P, wireless mesh networks, cooperation, clustering and BT is done, followed by presenting the main Python programming language features. Methodology, measurement setup and scenarios are explained in detail. Later on, the evaluation of the measurements is done by using Python software.

Lists of abbreviations, figures, tables and symbol are also presented at the beginning. In this report, citations are presented in the form "[reference]", where 'reference' can be seen in the bibliography.

MOBcom 07gr1112

May 2007
Aalborg University

Acknowledgement

We would like to express our gratitude to our supervisors Frank Fitzek and Gian Paolo Perrucci for their valuable guidance throughout this project. Their inquisitiveness and motivation helped us a lot in developing this report. We would also like to thank the people who directly or indirectly helped us to accomplish this project.

Beatrice Pietrarca   Giovanni Sasso

Chapter 1

Introduction

1.1 General Presentation of the Work

The main goal of this project is to investigate the possibility to benefit from cooperation by allowing the creation of a new architecture in wireless networks.

The idea is to give terminals the possibility to create local cooperating clustered networks within the existing centralized one. Terminals in each cluster can communicate among themselves by using a SR communication link, like in Figure 1.2 [1], as well as directly with the Access Point (AP), like in Figure 1.1 [1]. The information sent through the SR link is transmitted by using BT technology. The result is the creation of a P2P communication among terminals, i.e. mostly mobile devices, that use the BT technology to form mesh networks.

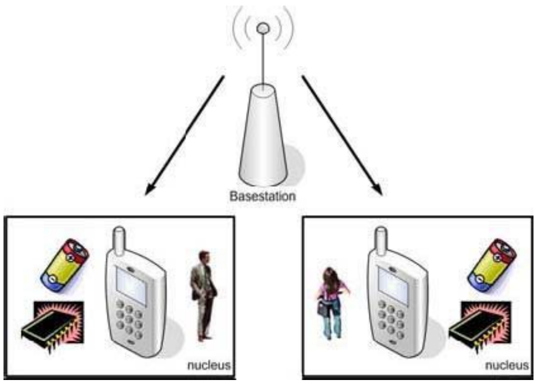

Figure 1.1: Communication involving the AP

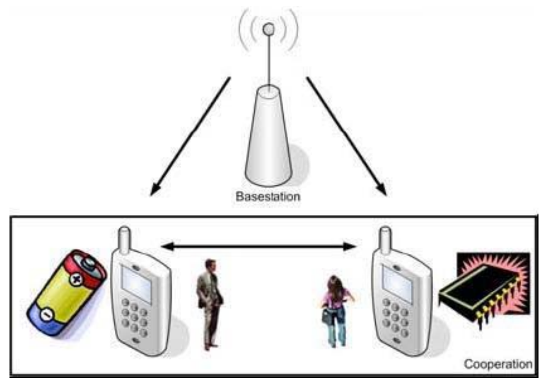

Figure 1.2: Distributed Cooperative Communication

This future architecture can provide better Quality of Service (QoS) in the network, and it can lead to both power and battery saving in the terminals, and bandwidth reuse, that are the most common problems regarding small and portable devices. In fact, terminals can use their cellular communication links to gather only partial service from the Base Station (BS), and then this partial service is exchanged within the cooperating cluster by using SR links.

Because this kind of cooperation involves mobile devices using SR BT links to form mesh networks, we obtained some measurements useful to determine statistics from the BT devices in the SR coverage. Measurements were taken in several places like buses, trains, shopping malls, bars, etc. Because of the large availability of mobiles devices supporting BT technology, a mobile phone equipped with BT has been used as sensing device. In particular, a N70 Nokia phone was collecting data. This kind of phone provides a flexible and powerful platform, i.e. Symbian OS (2^{nd} edition FP3). Symbian is an operating system built up for mobile devices, that can support programming languages like Python, Java, Symbian C++.

The phone was equipped with Python S60, so that by running a specifically designed Python script, it was possible to make measurements and supply information about the BT devices discovered in the area around. All Series 60 phones equipped with the Python interpreter are able to run the script, made available on our home page [2].

Data were collected first in the phone, then transferred to a Personal Computer (PC) on which Python 2.5 software was installed. By using that software, the analysis was provided.

Our focus on this project is to check the availability of BT devices as to form cooperation clusters in wireless mesh networks. Therefore, the work deals with results for mesh networks formed by mobile devices, like the number of nodes found, their types, and temporal considerations about the network.

After presenting the utilities and motivations of this work, in this chapter we will present some general statements regarding cooperation and clustering, and BT.

1.2 Motivations

Nowadays, BT technology has became one of the main point in P2P wireless domains. BT devices can be easily found: mobiles, laptop, etc. are only few examples that tell us how this technology is going to be used more and more often.

Till now, BT has been used at most for data exchange. But because of the potential availability of a large number and kinds of BT devices, they can be used for other purposes.

The idea is to be able to obtain cooperating clusters, i.e. cooperating terminals within each cluster in wireless mesh networks, in which terminals can communicate in P2P by using BT SR links.

1.2.1 Mesh Networks and P2P

A mesh network can be described as a cooperative wireless network in which nodes, each connected with several other nodes, are able to deliver and relay data. Thus, mesh networks employ multi-hop wireless forwarding in which communication between two end nodes is carried out through a certain number of intermediate nodes. The architecture is simple, because each node needs only to transmit to the next node. There are two different topologies of mesh networks:

- A "fully meshed" topology means that every node has a direct connection to every other node. This is a very elaborate and expensive architecture.
- A "partial meshed" topology means that nodes are connected to only some, not all, of the other nodes.

One of the most important characteristics is that the network infrastructure is decentralized, to avoid a central point of failure and control. Thus, services become more reliable. Then, this structure provides a dynamic self-organization and configuration of nodes, as well as a high scalability. In the last years, and in connection with wireless networks, the term "mesh" has been more often used as a synonym for "ad hoc" or "mobile" network. However, usually a mesh network can be (but does not have to be) a mobile network in which at least some of the nodes can change position over time. The most significant problems that affect wireless mesh include network scale issues, security and radio frequency interference.

Social networks or wireless grids are examples of mesh networks. In social networks, nodes use SR links to exchange data or mostly digital contents among them (See Section 1.3.4). Wireless grids are resource-sharing networks that connect devices with each others in a cooperative communication.

Mesh networks are the basis for any P2P communication.

The idea of P2P came out in 2000. Even the original Internet was modeled as a P2P system, then it growth more asymmetric [3].

In P2P networks nodes are connect via ad-hoc networks, so that communication does not involve a central server. This means that nodes are directly connected, as cooperating peers, without using the client-server scheme.

In fact, in a pure P2P network, peers, i.e. involved nodes, act as equal, merging the roles of both client and server, without the use of a central server. Each peer provides resources, such as bandwidth, computing power, storage space, etc. Thus, it is clear that as new nodes join the network, the total capacity of the system increases. That is not true in a common client-server system, where more users mean slower data transfer. P2P can use the existing bandwidth more efficiently.

Then, because in P2P networks data are distributed among nodes, the robustness of the system is guaranteed, also because without a single centralized server, a single point of failure is avoided.

As said before, P2P allows a decentralized application design, where nodes can connect directly each others. When nodes are scattered throughout a large area, so that direct contact between all of them is not possible, then the P2P topology can be replaced by a cluster tree topology [4]. More details about clustering will be discussed in Section 1.4.

P2P is generally employed in the case of file sharing, where it becomes especially useful. An example is the so-called Bit Torrent.

1.2.2 Bit Torrent

One protocol used to realize file-sharing is Bit torrent.

Bit torrent is a way of distributing a large amount of data in the network, so that clients can be able to access, request and transmit these data by using the Bit Torrent protocol. In order to share a file, a peer has first to create a torrent with information about the file and the tracker, i.e. the computer that coordinates the file distribution. In that way, when another peer wants to download that file, it has first to download the torrent file, so it can have the information from the tracker of which clients have pieces of the file. The file has been broken into small pieces, identically sized. A peer that has the initial file is called initial seeder, other peers that have copies of the same file are called seeders. Peers connected to the same torrent form a swarm. When a client wants to obtain a file, it has to connect to the peers in the corresponding swarm. If that swarm contains only the initial seeder, the client connects to it only and starts the downloading. If more seeders are present, in the moment in which a new seeder joins the swarm, it starts to download the file from another seeder, instead the initial one.

In this architecture, files can be exchanged according to a fair trading or randomly. In the first case, to send data a peer prefers the one that will send back data to it. In the second case, the peer is chosen randomly, it is not the preferred one. Thus, even better peers can be discovered and join the swarm.

A file-sharing system can work well in the case in which there is availability of multiple sources of the file shared, and the transfer speed is closely related to the number of peers sharing that file.

A big limitation is due to the fact that an identical copy of the file that has to be downloaded should be present in the network. An architecture like this is not able to use similar but non identical copies. The solution can be a system that provides a way to spot chunks of identical data. P2P file-sharing systems already split files into chunks typically based on byte length. Once the chunks are created, the system has to find the chunks sources for each chunk of the file. Then, because one identical chunk can appear in one file that is not identical but similar to the wanted one, an hybrid solution can be proposed. That is the idea of SET.

1.2.3 Similar-Enhanced Transfer (SET)

The idea behind SET works by identifying files that are similar to the one being downloaded, and finding identical pieces (chunks) within those similar files [5]. In this way it can accelerate the typical P2P downloading, because the potential sources of files being downloaded increase.

In this solution, similarities need to be detected.

The main problem of a file-shared system, in fact, is not that a source is not present on Internet, rather than it is present but in a different version. This is the case of a video in another language, or a song with a different or incorrect label. A common P2P system sees all these versions as different files, so it is not able to use them for the downloading.

SET gives a solution by comparing files that present similar sequences of data. And because it would mask similar but nonidentical files, it uses a process called Rabin fingerprinting [6]. This method sets chunk boundaries based on both size and properties of the data itself.

Once the file is split in chunks, a hash value can then be calculated for each chunk. This value can allow identical chunks to be recognized. Anyway, sharing and comparing all the hashes for the chunks of all available files can generate an enormous network traffic and wasting of resources. To avoid these problems, a scheme called handprinting has been used [6]. The calculated hashes are sorted lexicographically, and the first 30 of the resulting hashes are defined as the handprint of the file. Those handprints can then be sent to the global filesharing lookup table, allowing new clients to rapidly access them. The measure of the total similarities between two files can be done by comparing two handprints, so that if the similarity of a handprint is above a certain threshold, then the full list of chunk hashes is compared, and identical pieces of the files can be downloaded. In that case, only if the files are similar enough, a full comparison is done.

At least three parameters have to be set:

- Chunk size;
- The number of chunk hashes that go into a handprint;
- How to set the similarity threshold.

However, an increasing up to 500% of the transfering speed can be obtained.

1.3 Cooperation

The term cooperation holds the idea of "working-together" [1].

A group of people or entities that work on the same task can do it together in collaboration, rather than alone in competition. These entities have to share or give something in order to obtain a gain. Even if that is not always true, the main aim is that each entity is able to obtain a gain. Collaborating entities can form a unique virtual work entity.

Cooperation can be defined as a small scale or a long scale one. The first involves a few number of collaborating entities, while the second involves much more entities as to form a bigger collaborating group.

The idea of cooperation can be found in nature since long time ago. Nowadays, it has been developed in modern systems, such as cellular wireless communication networks. In that case, rather than realizing a higher centralized system, cooperation can led to a new architecture in which higher QoS, higher performance and less infrastructure costs and complexity can be reached.

1.3.1 Forms of Cooperation

The main idea behind cooperation is that each involved entity can obtain more by collaborating than it would by operating alone [1].

Several strategies for cooperation can be described.

A first example is the so-called selfish-cooperation. It originates in a scenario in which each entity tries to maximize its gain, so that it refuses to collaborate in the case in which cooperation does not give any clear benefits.

On the other side, the most generous form of cooperation is altruism. That is the case of one entity which is not interested on gain anything, but it makes possible to increase the others gains.

When each entity is focused on its own interests, a distinction between cooperation and defection should be done. Defection is when one entity stands alone rather than collaborating with the others because it sees no gain in cooperation. In the case in which, by collaborating, it is possible to have a greater benefit, collaboration is chosen.

For two or more collaborating entities, several theories have demonstrated that cooperation results as a better strategy respect to the case in which cooperation and defection are alternated.

1.3.2 Cooperation in Wireless Networks

Traditional cellular communication systems are based on the transmission of data between BSs and one or more terminals (Mobile Station (MS)). A new step towards the future communication in wireless networks is the idea of forming local distributed networks, such as mobile ad-hoc or P2P networks.

This can be possible by enabling mobiles to communicate directly each others by using a SR communication technology (BT, Wireless LAN (WLAN), etc.) with higher data rates. In this case, even if the cellular communication link between the BS and one MS is not established, a MS that is not directly connected to the BS can use another MS as relay. This idea the so-called relaying cooperation. The only need is that the node is available and ready to cooperate and forward packets for other nodes. In other words, a node has to forward packets for other nodes in order to make possible multi-hop communication.

In general, cooperation in a wireless scenario means an exchange of information and data over SR links. This leads to a better use of the spectrum and a reduction of the infrastructure costs.

Some properties common to all cooperative communication systems can be defined as follows [1]:

- Cooperative systems are highly distributed, not centralized;
- Users in cooperative systems can share resources (such as bandwidth, battery power, etc.) as to form a common pool of resources to achieve common goals;
- By using pooled resources, higher advantages can be obtained compared to the case in which individual resources are used;
- These pooled resources have to be accessible by the users without commitments, otherwise a consumption in terms of bandwidth or battery is in general obtained.

1.3.3 Motivations for Cooperation

Cooperation makes sense if it is possible to have a gain by doing it.

One of the most common problems for devices in a wireless network is their limitation in battery energy, capacity and computational power. Two or more near mobile devices can form together a cooperative cluster as a micro network in the centralized cellular network. In that case, while typical distances between the BS and the MS cover several kilometers, the distances between the devices in a cluster are around tens of meters, so that it is possible to refer to them as SR links.

The final result is a hybrid network from the combination of a cellular and ad-hoc network. It is possible then to achieve a better use of resources, mostly energy and spectrum, and to improve the system overall performance.

In general, two different metrics for cooperation can be stated:

- Physical resources saving (energy, etc.);
- Communication domain resources saving (bit rate, latency, etc.).

In a such described scenario, the BS can send data using the high power and low data rate cellular link. Then, these data can pass through the devices in the cluster by using the low power and high data rate SR link.

The advantage is clear just considering a simple example, like the video streaming or file downloading. The main idea can be to use each terminal to receive a portion of the whole stream via the cellular link and then pass it on over the SR to the other terminals of the cluster. The gains are in terms of power consumption and general costs. For example, the costs of the service can be shared among all the terminals in the cluster. On the other hand, a better spectral efficiency is obtained. That is because an unfortunate loss of a packet will need a retransmission just over the SR, i.e. locally, and not from the BS. Information is spread over the cellular network and distributed in the cluster, and only at the end combined at the destination. This makes possible a higher security level in the system.

Beyond the cellular link between the BS and the MS, that can be defined by Universal Mobile Telecommunication System (UMTS)/Global System for Mobile Communications (GSM) standards, the SR links between cooperation devices does not need to be defined.

In this project the possibility of being able to establish SR links via BT is taken into account.

Once the cellular link is established, the cluster should be defined by identifying the cooperative entities. That is not really easy in a moving scenario. Here, entities can stay together for long time if they are moving around the same area or toward the same direction, or, of course, if they are standing in the same place. But, they can also just pass by and then stand alone or join continuously different clusters for most of the time.

As the scenario changes, new cooperative groups need to be formed.

In particular, dynamic cooperation systems have to be carefully designed in order to avoid unstable situations and fluctuations of the QoS. In this case, the system should be able to adapt changes.

1.3.4 Cooperation based on Authentication: Social Networks

A mechanism of authentication between nodes, and the possibility to create a cooperative architecture based on reciprocity between recognizable nodes requires that each terminal is able to identify the others in order to decide whether or not to cooperate.

This is related to a particular kind of network called social network on Internet, already cited in Section 1.2.1.

The idea is that an initial set of founders sends out messages inviting members of their own personal networks to join. So, if the new members repeat the process, the number of members and links in the network increases. Most of connections are with people who are nearby. However, the *Six degrees of separation* tells that, if a person is one ”step” away from each person he or she knows and two ”steps” away from each person who is known by one of the people he or she knows, then everyone is no more than six ”steps” away from each person on the Earth. In this way, because in a social network everybody can invite his/her friends to join the network and each friend can do the same, it can be possible in a moderate number of steps to cover the whole area of the world.

In this direction, some applications for Nokia mobile phones have already been created. Examples are Nokia Sensor, or Smartex.

Nokia Sensor uses BT technology to sense the other sensor phones nearby, and connect with them to see their profiles and send them messages. The advantage is that this kind of communication is for free, because phones running the Sensor application communicate directly without going through an operator network service [7]. More details can be found in [8].

Smartex have been created in order to allow terminals to exchange digital content between them using the SR communication, based on BT, under the control of the cellular network [9]. These digital contents can be, for example, properties of the mobile terminals, as ring tones, or collections, as baseball cards, or for multi player role games, i.e. virtual tools. Initially, the BT search for new neighbors is done. Each user can configure its wish list and its trade list, i.e. the contents that it is looking for and the ones it wants to exchange. The application will automatically match wish and trade lists of users. More details about this application can be found in [9].

1.4 Clustering

Clustering is a way of grouping together devices in wireless ad-hoc and sensor networks. In this way, it is possible to obtain energy saving in the communication among nodes, by creating a hierarchical structure. This technique is especially efficient in networks with a large number of nodes, where scalability is really important and the routing problem needs efficient distributed algorithms.

1.4.1 Wireless Networks: Devices Limitations

A network includes a set of devices. The simplest idea of device is a common mobile phone, in a mobile wireless network. Each device has, in general, limited resources. First of all, they are limited in battery, and, due to limited battery life, they suffer from severe energy budget. Then, they usually suffer of low memory, little computing capability, very low data rates, low bandwidth processing, variable link quality, etc. [10].

The clustering mechanism can guarantee a better communication scheme, letting a more efficient use of the critical resources located in the wireless nodes, without decreasing the communication performance.

1.4.2 Clustering Architecture

Clustered networks are based on the organization of devices in a hierarchical mode, obtained by separating them into groups called clusters. Each cluster is established with P2P capability. Within the cluster, not all the devices have the same role, even because in a non-homogeneous cluster not all the devices have the same hardware and the same functionalities. They can vary in size, communication capabilities, computational power, memory, mobility and channel capacity [11].

Three main elements can be defined in a clustered network [10]:

- Base Station (BS);
- Common nodes or cluster members;
- Cluster Head (CH).

Generally, in a heterogeneous network, the CH has more energy and a better hardware than common nodes, while in a homogeneous network the CH is similar to normal nodes.

The scheme guarantees the communication between the BS and any node (or between nodes in different clusters) by using the CH. Therefore, the CH is a sort of gateway between the nodes and the BS.

Practically, two different kinds of communication can be possible in a scheme like this: intra-cluster and inter-cluster communication.

The first interests the communication within the cluster of nodes among themselves and with the CH. The second involves the communication between the CH and the BS, or among the CHs.

The idea is to allow only the CH to send data outside the cluster, for both the BS and other CHs. Thus, if a node in the cluster wants to send data outside, he has to send the information to the CH elected. In this way, the CH can not only generate data, but also collect the information from normal nodes and transfer them to the next node [12]. The CH aggregates data and sends them to the BS.

Thus, the communication is based on a Store & Forward mechanism, because the message passes node through node until it reaches the destination.

Single-hop or multi-hop communication can be used within a cluster to reach the cluster nodes. New devices can join the cluster. If the cluster varies very fast, the performance can be reduced because of a large amount of control message signalling that can damage the devices limited power.

1.4.3 Cluster Establishment and Advantages

During the cluster creation, some parameters have to be chosen. First of all, the cluster size and form. Then, an algorithm for the election of the CH is needed, as well as general rules to avoid collision and guarantee the energy saving.

This is not a very simple task.

Anyway, a communication based on clustering can benefit from several advantages.

As said before, clustering is especially useful in a several nodes communication, i.e. in a scenario that needs scalability. In this case, clustering leads to more efficient resource allocation. In fact, it is only the CH that can send information outside the cluster, so that all the nodes within the cluster do not have to share the channel with the ones within another cluster. Therefore, a spatial reuse of the bandwidth is possible, and this increases the system capacity. Because only the CH performs the data aggregation, and not all the cluster nodes, they can save energy.

Data aggregation also helps to eliminate redundancy, so the overall data sent to the BS is reduced.

Latency is also reduced, because routing is based on clustering and the total information in the network is reduced, therefore delays are reduced.

One of the most clear advantage is in term of energy saved. Nodes spends energy to transmit, receive and eventually relaying information. In this direction, clustering tries to solve the so-called maximum lifetime problem. This means that, given an initial energy, the hope is to be able to provide data updates for as long as possible.

1.5 Bluetooth

BT is a short-range digital wireless technology that replaces cables in communication.

In the last years, in fact, wireless technologies are rapidly developing according to the need for mobile and dynamic communications. For that reason, cables need to be replaced with wireless links.

Wireless means communication without wired, i.e. without any cables. In small geographic range, WLAN can be implemented to connect several devices with high speed.

BT carries the WLAN concept to a smaller scale, usually around 10 meters range. It can be useful to connect very close devices or devices on a person. That concept is called Personal Area Network (PAN).

Via BT, cooperation and resource sharing between various types of equipments can be realized with low power consumption and low cost in SR area.

Before explaining the BT technology in more details, it is useful to give some reasons why wireless technology is so useful in the modern world.

1.5.1 Wireless and Wired

Wired connection means a little mobility, while wireless allows devices to remain in communication even if they are moving. Together with mobility, wireless can give cost savings, because no cables need to be installed or upgraded to make available the connection. In fact, the only physical medium needed is the air, so connections are more versatile.

Anyway, some disadvantages arise from the use of wireless technology. First of all, the wireless communication meets a significant attenuation in the terminals communication while moving. For that reason, a possible collision between signals takes much more time to be discovered. This means delays in transmission and loss of communication efficiency [4]. Bit Error Rate (BER) is higher in wireless, so more control bits are needed. The BER reduction is possible by transmitting with lower data rates, so that communication will result less fast. Then, security problems come because signals are transmitted directly in the air, so they are easier to be intercepted.

BT has incorporated several techniques to alleviate these problems [4].

1.5.2 Basic Bluetooth characteristics

BT is a short-range communication technology that permits to connect and interact with different devices such as phones, laptops, PDAs, etc. without a physic cable connection, and transmit data at a rate of about 720 Kbps .

As a cable-replacement technology used to connect several near devices, BT has created the notion of PAN, that can be seen as a close range wireless network [13]. Because it uses radio links and radio waves, this technology belongs to the group of wireless technologies. The advantage is that radio is not directional, so no Line Of Sight (LOS) is needed between connected devices.

The main characteristics of BT technology are robustness, low power, low cost, security, ease, no cost to use it and small size of the chip-set. Under these conditions, connectivity becomes simple and seamless [13].

BT technology started from 1994, but only in 1999 Version 1.0 came out. From November 2004 Version 2.0 was specified, compatible with all the previous ones. The latter can support the enhancement of data transfer and a faster speed in transmission (up to three times), with a lower battery consumption due to a reduced duty-cycle and a lower BER. Then, because of a larger available bandwidth, the multi-link scenario is simplified [14]. The low power consumption, small size and low cost of the chipset solution enables BT technology to be used in the thinnest of devices [14].

BT operates in an ad-hoc fashion [3], where links can be established among in-range devices.

The BT transceiver operates at the frequency of 2.4 GHz. The reason for that choice is that the 2.4 GHz is the only band that is mostly allocated worldwide and requires no license to be used, it's unlicensed [4].

1.5.3 BT Device Address and Name

Each BT device can be uniquely identified by a Media Access Control (MAC) address known as the BT Device Address (BT_ADDR), assigned by IEEE [13]. This is specified by 48 bits, that are divided into three parts [4]:

- LAP (Lower Address Part) consisting of 24 bits;
- UAP (Upper Address Part) consisting of 4 bits;
- NAP (Non significant Address Part) consisting of 16 bits.

The LAP and UAP are used to identify the piconet, determine access codes and set the frequency hop channel. The NAP helps make the address unique for each device and it is also useful for security problems.

The BT address is permanent, even if the BT name changes. The name can be assigned by the user. Its maximum length is 248 bytes, encoded into the ASCII 8 bit-format [4].

1.5.4 Piconet

Each BT device has its own and unique BT address, as said before, and a BT clock.

Devices can belong to two categories: master and slaves.

One BT device can connect with one or more other devices by creating a SR network called piconet. Every piconet can contain at most 8 devices, and each of them can belong to different piconets simultaneously. Two piconets can communicate in order to create a bigger network, called scatternet. The scatternet is a particular kind of BT ad-hoc network.

The BT wireless specifications allow to develop devices for both link layer and application layer, which can support both data and voice transmission differently to many other standard wireless networks [14].

BT technology uses Adaptive Frequency Hopping (AFH) to reduce interference at 2.4 GHz, due to other signals transmitted, and fading, due to the continuous movement among devices. All the devices that belong to the same piconet are synchronized each other with a common clock, that is just the BT clock of one device known as the master, and all the other devices, known as slaves, use it as a synchronization reference.

In each piconet there is a physical link between the master and all the slaves, but the slave are not directly connected each other with physical links. The total time is divided into time units called slots, because collision has to be avoided. Therefore, data is transmitted in packets using these time slots, and a packet can be allocated in several consecutive slots.

1.5.5 Classes of devices

The BT specifications define three different classes of devices, each class has a different range cover:

- CLASS 3- the cover range is 1 m
- CLASS 2- the cover range is 10 m
- CLASS 1- the cover range is 100 m

There is also a minimum range for a BT radio link, that is about 10 cm. Generally a smart-phone implements CLASS 2, while CLASS 3 is mainly used in industry branch.

1.5.6 Security

BT has a built in security due to a 128 bit encryption and a PIN code authentication made in the first connection. Once connected, devices can exchange data with safety.

Multiple access data is transmitted among the devices with Time Division Multiple Access (TDMA), so the information is divided in several packets positioned in time slots. The use of Time Division Duplex (TDD) enables the full-duplex transmission.

1.5.7 BT discovering

To discover other devices in the neighborhood, BT use the inquiry procedure.

This is an asymmetrical procedure: a BT enabled device try to find nearby devices sending periodic inquiry requests, and each BT enable device that is able to be found replies with inquiry responses. These inquiry requests and responses are sent through a special physical channel. Finally, after the first inquiry procedure when all the enable devices are physically connected to each other within a piconet (connected mode), it is possible to create additional logical link. In this case, it also possible to be connected to more than one piconet [14].

1.5.8 Applications

Because BT technology replaces cables and produces a low-cost, low power, reliable and short range communications links, it suits to be used for connections between a large variety of mobile devices [13], such as:

- Mobile phones to Public Switched Telephone Network (PSTN) through AP;
- Mobile phones to computers;
- Mobile phones to headsets;
- LAN AP for laptops or palmtops;
- Notebook, palmtop or others Internet access devices via a PSTN AP or module;
- Communication between laptops and palmtops.

BT is now available in a vast range of applications, from mobile phones to automobiles to medical devices, and many more.

Chapter 2

Measurement Testbed

After a general introduction about the project in the previous chapter, here we are going to present the work regarding the measurement part.

This chapter will throw light on the measurements and the way they were obtained, as well as how to use them to evaluate mesh networks.

Some general information about Python are given in Section 2.1. Section 2.2 describes in details the measurement setup and how data are obtained and collected. The scenario under assumption in the measurement campaign is described in Section 2.3. Section 2.4 deals with boundaries and limitation problems by using BT technology. The methodology is described in Section 2.5.

2.1 Python

Python is a portable, interpreted, object-oriented programming language.

It is both robust and uncomplicated, since it has inherited both the power of traditional complex languages and the easy-of-use of simple scripting and interpreted languages [15]. It is freeware, so there are no restrictions on copying or distributing it.

Python development begun in late 1989 in Netherlands by Guido von Rossum as a really innovative language. The elegant, simple and easy-to-learn syntax is built around a small number of high-level types like Python's lists (resizable arrays) and dictionaries (hash tables), together with a simple structure.

The syntax lacks of usual symbols for accessing variables or code blocking definitions such ad dollar ($), semicolons (;) etc., that are used in other more complicated languages. Thus, its code is much more easy to read.

Python is also commonly used in graphical user interfaces, internet scripting and database programming. Together with the clear syntax and coherent design, Python is built up around an extendible architecture and a support for object-oriented development and powerful programming constructs. It can be extended by adding new modules implemented in compiled languages such as C or C++. Such extension modules can define new functions and variables as well as new object types [16]. For this reason, Python has an object-oriented nature: a programme can be built up with different modules and modules can define together classes.

This leads to the creation of a modular architecture and code reuse in order to obtain flexibility, consistency as well as faster development time. In fact, the code can be separated into multiple files or modules. Each of them can be accessed according to the same syntax, either if it belongs to the Python standard library or if it has been created and developed by an user.

It can be stated that, in Python, all code exists on modules, which import services from other modules; these modules contains statements which operate as objects.

A module is the biggest programme unit, which imports modules.

Statements exist within a module; examples are assignments (=), tests (if), loops (for, while). Some of them are used to create new objects. For example *def* is used to create a function, *class* to create a class, and *try/exept* are used to manage exceptions.

Indentation is used to separate statements, and objects are processed by statements. Python modules are executed in their statements when they are imported, or they can run as scripts [15].

A large number of extension modules have been developed for Python, as well as a numerous external libraries. Some of these libraries include: networking, Internet/Web/CGI, graphics and graphical user interface (GUI) development, imaging, numerical computation and analysis, database access, hypertext (HTML, XML, SGML,etc.), operating systems extensions, audio/visual, programming tools, and many more [15]. Python extensions can be written in C and C++ for CPython, and in Java for JPython [15].

Python lacks of variable declarations, because variable come to existence by assigning values to them, and it lacks of block markers and boundaries, that are created automatically.

Python implementation is available in a wide number of platforms, so that it can be easily defined as portable.

The debug cycle is really fast because there is no compilation step. A bug or bad input will never cause a segmentation fault, but Python supports exceptions when errors occur, i.e. when the interpreter discovers an error, an exception is generated [17]. When the program doesn't catch the exception, the interpreter prints a stack trace. The stack trace will indicate the name, location and type of the exception. The exception handler is able to recognize the exception and act in the appropriate way.

In addition to that, then, the debugger is written in Python itself.

Also the memory management is performed by the Python interpreter, so that the application will be more robust and it will take less development time.

Even if Python is classified as an interpreted language, actually it is a solution between interpreted and compiled languages, the so called byte-compiled [17]. This leads to the increase of the performance.

Python is available on several platforms, such as UNIX, Windows, OS/2, Mac, Amiga, and many others.

There are also several available versions of Python. In this project we used Python 2.5 in the computer, and Python S60 in the Nokia mobile phone.

2.1.1 Python 2.5

Python 2.5 is the latest major release of Python, with language and library improvements. as regards the library improvements, new modules have been added, such as ElementTree for XML processing or the *ctypes* module for calling C functions [16].

On the other hand, as regards the language, conditional expressions were added using a novel syntax, some cases of exception handling are handled better and new statements have been added.

2.1.2 Summary

The reasons that make Python a powerful, but simple programming language can be summarized in its following characteristics [18]:

- Object-oriented programming;
- Embedding and extending in C;
- High level data types and operations;
- Classes, modules and exceptions;
- No type declarations;
- No compile or link step;
- Large set of data types:
- Freeware;
- Compatible for almost all the platforms.

  According to that, Python results as an easy to learn and rapid development language.

2.2 Measurement Part

Data are obtained by running a Python S60 script installed on a Nokia N70 phone.

The Python script collects information about the BT devices found in its coverage while running.

Once the phone is switched on and the application is launched by choosing the *Start* option in the menu, a list of possible places appears (Figures 2.1 and 2.2). Therefore, it is possible to choose an existing place in the list, or to insert a new one not already present. In this way, the place in which the measurements are done is specified.

After the position, the time information is taken. It contains date, day and time (hh:mm:ss).

Then, the GSM position is also taken. The GSM information obtained specifies the Mobile Country Code, Mobile Network Code, Location Area Code, and Cell ID. In this way, a first identification of the location in the network into which the phone is logged is given.

After that, the programme starts to scan how many BT devices are in the sensing phone BT range (10 meters). For each device, information about address, name, Major Service Class (MaSCl), Major Device Class (MaDCl) and Minor Device Class (MiDCl) are obtained. For more details about Service and Device Class, see Appendix B.

The scanning is repeated with a frequency of 30 seconds until the *Stop* option is chosen (Figure 2.4). The pause interval is set to 30 seconds, first of all for problems regarding the cache refreshing. By setting a smaller interval, the cache has not time to be refreshed. Therefore, it is possible that devices found in the previous scan, but not present anymore in the BT range, are still in the phone cache and are given as wrongly discovered devices. On the other hand, an interval of at least 20 seconds is recommended in order to save the battery consumption in the local device, and in order to minimize interference with other wireless networks and BT traffic.

After each scan, the information obtained are stored in a XML file, that represents a useful way to collect and store data (see Appendix A). A different file is created each time the *Start* option is chosen, i.e. each time a new scan is done. Then, within this search, for each scan the information is saved in the file. The XML file is closed once the *Stop* option is chosen and the file is then stored in the phone. By choosing *Show Log* the XML file is shown on the mobile screen. By choosing *Send File* in the menu (Figure 2.4), the file can be also sent either on Internet by using a General Packet Radio Service (GPRS) connection or to another device over BT. An example of the XML file structure is available on our web page [2].

It is also possible to show a bar graph on the mobile screen, as in Figure 2.3. The histogram is a graphical representation of the new and the old devices discovered in the last scans. For each search, the height of the blue bar is proportional to the total number of devices found, while the height of the red one is related to the number of those devices which were not present in the previous search, i.e. the new discovered devices. As given by Figure 2.3, during the first search, the two bars have exactly the same height because the devices discovered at the beginning are all new.

Figure 2.1: Screenshot Menu Start

Figure 2.2: Screenshot Places

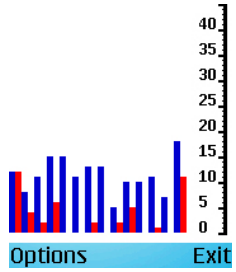

Figure 2.3: Screenshot Graph

Figure 2.4: Screenshot Menu Stop

2.3 Measurement Scenario

This section describes the scenarios in which the measurements were carried out. The places in questions can be divided in two main categories:

- **Lowly dynamic places**: Places like bus, train, metro, bar, university kantine, etc. are indoor places in which there is no possibility to change position very often. Thus, always the same people are met several times, and the number of new devices is not too big. People can stay together for long time (chatting, waiting or drinking).

- **Highly dynamic places**: Places like downtown, shopping malls, station, etc. are crowded of people that are always moving, and even the sensing device is moving from time to time, so that we expect to have a large number of new devices in each search. In particular, downtown is the scenario with the highest degree of mobility.

Another scenario presented is a big center (Gigantium) during a technological conference. There were some stands inside the building. Thus, the person carrying the sensor device was moving and this place can belong to the second category above described. However, while some discovered devices were fixed (this is the case of laptops, computers, peripheral devices presents in the stands), usually visitors were carrying their own devices, especially mobiles.

In all the places cited above, the sensing mobile has the possibility to move as it is carried by a user. A different category can be defined in the case in which, even if people around are always moving, the sensing phone is fixed. An example is represented by the measurements taken in the airport. In fact, a set of measurements was taken during ten days in Aalborg airport. One N70 Nokia mobile phone was boxed and put on the wall at the height of around two meters. By running the Python script, the phone was able to collect information of the BT devices present in the waiting room of the airport, everyday from 5 o'clock in the morning till midnight. Due to security reasons, the device was placed at the boundaries of the waiting room. Thus, it will have only find half of the devices possible compared to a device placed taking measurement in the middle of the room.

2.4 Boundaries: Limitations in Number of Devices

One of the main limitations in the idea of a cooperating clustering wireless network using BT SR is the BT itself.

The cooperation is among BT devices, and nowadays most of phones, laptops, etc. are equipped with this technology. However, the problem is that many users do not find useful to switch on the BT in their devices. In fact, leaving on the BT means a cost in terms of battery mainly, even if BT was designed to require as little power as possible. Thus, because many people tend to switch off their BT, it is clear that the number of BT devices found is smaller than the one that could be possible if all the available BT are switched on.

Then, another possible reason is related with privacy, as well as security problems, because BT uses radio waves for communication. Data are sent without any cable supports, so that they can be easily intercepted by third parties and even attacked by viruses. Hackers can monitor activities and access data that are stored in a BT device. To avoid these problems, a user can set its device as non-discoverable while its BT is not being used. In that way, this device becomes useless for cooperation.

As explain in Section 1.5.7, the BT discovering procedure requires inquiry requests and responses. Thus, the procedure takes some time to connect physically all the devices.

BT offers a range of typically 10m. Some versions of BT can support much greater ranges (up to 100m), but these requires more power. Therefore, because the range cover is quite small, it is possible that, even if there are some available BT devices, they are not in-range, so they cannot be useful.

2.5 Methodology

The measurements taken as explained in Section 2.2 are stored in a XML file. As new data are obtained, the XML file is updated, so that the whole set of measurements is stored in a common source to be then analyzed. The analysis interests the number of devices discovered, comprising both the total devices found and just the new ones. Among the total devices, the device type is checked, and among phones, different kinds of phones are distinguished, as will be described in Section 3.3. Some statistical and probability considerations can be done by analyzing the Probability Density Function (PDF) and Cumulative Distribution Function (CDF) of the number of devices obtained (Section 2.5.4), as well as the confidence interval (Section 2.5.3). Then, the connection time in which it is possible to benefit from cooperation can be calculated, as in Section 2.5.5. It is the time in which the cluster can be kept alive. Some considerations about the scanning time required for a certain number of devices will be presented in Section 2.5.6.

2.5.1 Number of Devices and Time of Connection

Number of devices means not only the total number of devices found, but also the number of new devices obtained. These data are collected together in a bar graph, with different bars for total and new devices for each search.

Then, because the final idea is to obtain a clustered cooperating network, it is important to analyze how many devices are together and for how long, so that a cluster can be formed.

In order to create a good cluster, at minimum three devices are needed. The cluster becomes more stronger as the devices involved are together for long time, i.e. they are the same and do not change very rapidly. In fact, if the cluster changes very often, a lot of power will be spent for signalling.

For that scope, for each search the different devices found are presented. One device can be univocally specified with its Bluetooth Device Address (BT ADDR), as already described in Section 1.5.3. If it is possible to recognize a BT device from its BT ADDR, it is also possible to identify it every time one device wants to start cooperating. This means that a cooperation based on authentication of nodes can be done, because the nodes are able to recognize themselves. Figure 2.5 shows two different examples of possible devices found in a range of four measurements.

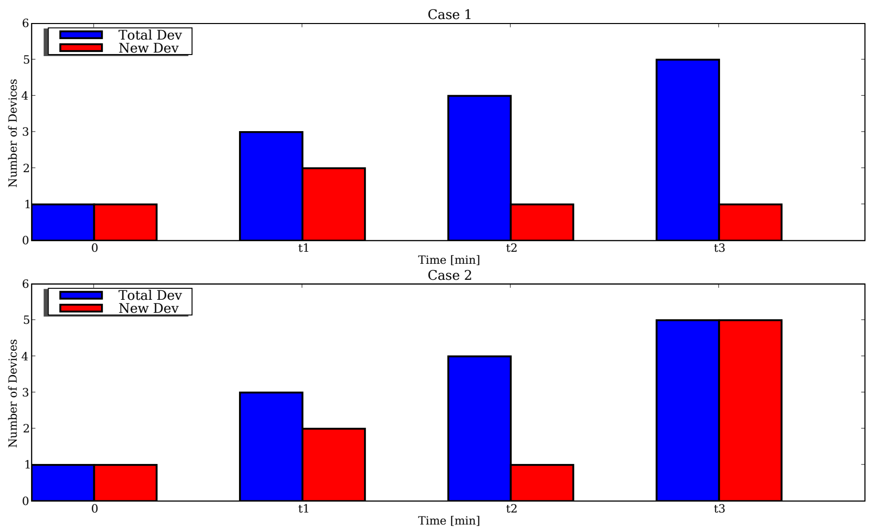

Figure 2.5: Different Signalling Situations

The histograms are created looking at the number of total devices (blue bar) and the new ones (red bar) for each search.

The first three measurements are equal in number of both total and new devices found, the fourth one has a different number of new devices found, even if the total number is the same. Let's define the cluster as a group of at least three devices. It can be seen that, till t1, a cluster cannot be created since for the first measurement we have just one device. After this time it is possible to create a cluster. In fact, from the second search the devices discovered are equal or more than three, so the cooperating group can be defined.

As clear from definition, during the first search red and blue bars have the same height because all the devices discovered are new.

Then, the number of tot devices increases of two units, that are the only new ones because the red bar has height equal to two. During the third search, in four devices only two are new and the other two were still present in the previous search. Thus, the signalling on keeping the cluster is not too high. It is possible that this new devices join the cluster, so some parameters have to be set and some power and bandwidth will be used. Otherwise, it is possible to keep a three unit cluster, i.e. the old one, and do not let the new device to benefit from cooperation.

The fourth search is completely different in the two cases shown in Figure 2.5.

Even if in both cases it is possible to have a cluster, because the number of devices found is bigger than three, there is a clear difference in terms of signalling. In fact, while in the upper subplot, defined as Case 1, the cluster can remain exactly the same as before established, in the other subplot, i.e. Case 2, a totally new cluster has to be defined. In Case 1 there's only one new device discovered, and it is possible to let it join the cluster, spending some resources, or simply the cluster can remain the same as before. In Case 2, the cluster can be still obtained, but now all the devices found are new. It means that some resources have to be spent to create the cluster from the beginning, because no one of the devices was already present before. If at least three devices were old, the problem could have been solved just keeping the previous cluster, without considering the new available devices. Anyway, it is possible that these new devices are present in better conditions of power or resources, or are closer to the target one, so considering these new ones instead of the devices already present can help on achieving a better cooperating cluster.

Thus, we can conclude that the red bar has a meaning in terms of signalling, i.e. compared with the height of the previous blue one, its height gives information of how many resources have to be spent to create the cluster or modify an existing one.

The x-axis shows the times in which each measurement starts. By calculating the interval of time in which the number of devices found is greater than three, the time of connection is given. By referring to this simple definition, from Figure 2.5 the time of connection is equal to $t3-t1$. More details regarding different types of connection times are defined in Section 2.5.5.

2.5.2 Types of Devices

The analysis of the devices type can be done by looking at the MaDCl. In this way, it can be set if the BT device is a phone, a computer, etc. The specific type within a category can be found by looking at the MiDCl (See Appendix **??**). This is useful because our aim is to establish P2P networks among mobile devices, so that it is very important to understand the availability of mobiles as BT devices.

2.5.3 Confidence Interval

The confidence interval, also referred as HL, is an interval in which a measurement falls corresponding to a given probability. Usually, the confidence interval of interest is symmetrically placed around the mean value obtained from the whole set of measurements. For a population whose distribution is normal and whose mean μ is based on a random sample of size n, the confidence interval is:

$$HL_{95} = \mu \pm z^* \frac{\sigma}{\sqrt{n}} \tag{2.1}$$

where 95 refers to a probability of 0.95.

Anyway, for a common distribution, if n is big enough, the error by approximating it with a normal distribution is small.

The value z^* represents the point on the standard normal density curve such as the probability of observing a value between $-z^*$ and z^* gives the desired confidence interval.

In this section, the confidence interval considered is equal to 0.95. This means that the calculated interval will correspond to the probability of 95% of values obtained falling within that interval, that is centered around the mean value. In that case, $z^* = 1.96$, so that

$$HL_{95} = \mu \pm 1.96 \frac{\sigma}{\sqrt{n}} \tag{2.2}$$

2.5.4 Probability Distributions

Let's consider the number of devices as a discrete random variable X, that assumes values x in the alphabet $1, ..., N$, where N is the maximum number of devices found.

The probability distribution of this random variable can be described in terms of PDF, CDF and Inverse Cumulative Distribution Function (ICDF).

For a possible number x, the PDF of X is given by [19]:

$$x \longrightarrow f_X(x) = P(X = x) \tag{2.3}$$

It can be seen as the probability that X is equal to a possible realization x.

For a possible number x, the CDF of X is given by:

$$x \longrightarrow F_X(x) = P(X \leq x) \tag{2.4}$$

In other words, it is the probability that the variable X assumes a value less than or equal to x. The CDF is monotone increasing and its values are between 0 and 1.

The ICDF is defined as 1−CDF. Thus, it can be seen as the probability that X is greater than x.

2.5.5 Duration of Cooperation

Three different definitions of time of cooperation can be given. The simplest one, referred as Coop1, is the time in which three or more devices are in the range of the sensing mobile, and it doesn't matter which ones, they can change or remain the same ones. This time is referred as T_{coop1}.

This initial idea can be elaborated. Let's fix a cluster of three devices around the sensing one, and they are chosen randomly among all the available devices. The cooperating time can be defined looking at how long this cluster can survive. This time T_{coop} can be calculated in two ways:

- In Coop2, T_{coop2} tells how long at least one of the three devices that creates the initial cluster is still in the cluster. As soon as one device in the initial cluster is not present anymore, it is replaced with an available one, chosen randomly.
- In Coop3, T_{coop3} tells how long all of the three devices that creates the initial cluster are in the cluster all together.

In both cases, the cluster cannot be created in the case in which less than three devices are discovered, so we have to wait until at least three devices are present, if more only three will be chosen randomly. Then, in Coop2, even if one or two of the initial devices are still present, but they are the only ones, so the total number of devices is less than three, the cluster stops to exist. From definition, T_{coop2} and T_{coop3} are usually smaller intervals in respect to T_{coop1}, because Coop1 just needs at least three devices to establish the cluster, it doesn't matter which are them, while in the other cases there is a constraint about the devices that have to form the cluster, as explained above. We are not considering what can happen between one search and the one just following. In fact, as said before, there is a pause time of 30 seconds between one search and another. Within this interval one device can go away, and then come back, but the programme cannot manage that situation, and nothing will appear in the measurements.

2.5.6 Scanning Duration

The analysis has been also interested in the time employed to scan the devices in the BT coverage, i.e. the scanning time vs the number of devices found. The scanning is restarted every 30 seconds. However a scan takes longer (because of a large number of surrounding devices), so the analysis deal with the evaluation of the scanning period. We use this result of our measurement testbed to understand how fast the sensing device can get an understanding of its surrounding.

2.6 Server Uploading and Real-Time Post-Processing

The measurements carried out during the overall project and the results obtained are available on our web site [2]. A server Apache is always listening on a specified port. As soon as a new measurement is done, it is possible to send the XML file to this server from the sensing mobile, in which a client program is running. The file is sent by using a 1acGPRS connection. In this moment, the home page of our web site is automatically updated with a new link in the menu. This link addresses to a page in which new figures regarding the file just sent, and the file itself are present. In fact, when the server receives the XML file, data are stored on its hard disk, and, because it is equipped with a Python 2.5 interpreter, automatically the Python program is launched, providing all the results and figures. Each measurement (XML file and figures) is stored into a folder and addressed by its link. Figures are the same as will be presented and analyzed in Chapter 3 of the report.

Chapter 3

Results

In this chapter, we present the results regarding the measurements taken during the overall period of the project in different places.

First of all, analysis settings are defined in Section 3.1. Then the results for each metric defined are presented in sections 3.2, 3.3, 3.4, 3.5 and 3.6. Section 3.7 deals with the results obtained in a set of 10 days measurements taken in Aalborg airport.

3.1 Analysis Settings

Although the measurements were taken in several places, the results will be presented just for the most significant ones. The analysis of the information stored in the XML file is done with Python 2.5 software installed on a computer. All the plots are realized in Python, by using Matplotlib [20]. This is is a python 2D plotting library that, used in python scripts, helps makes possible to generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc.

3.2 Statistics about the Number of Devices Discovered

The first part of the analysis focuses on the number of devices found in each search. Figure 3.1 shows some results about the devices found in the Student House, i.e. an indoor bar where students usually meet. In the upper plot the relative devices with their BT ADDRs found in each search. Different colors are chosen to represent different devices. Because the availability of colors is not so big, it can be seen that one and only one color is associated to one BT ADDR, but the same color can correspond to more than one BT ADDR. However, even if two or more devices have the same color associated, they will lay in two differen rows of the plot, in order to avoid confusion. In fact, each line (y axis) refers to one BT ADDR, i.e. one particular device. The x axis represents the indices of measurements taken. If the device is found in the $i-th$ search, the rectangle corresponding to the crossing of the y axis line for its BT ADDR and the $i-th$ x axis line, i.e. the $i-th$ measurement, is filled with the color associated to its address (See Figure 3.1). The order in which the different BT ADDRs are put in the y axis is proportional to the frequency in which the device is seen. This means that, by looking from the bottom to the top of the plot, the devices are seen more and more rarely. In other words, the first line down corresponds to the device more frequently seen; then the top is for the device more rarely met. Even if there are devices seen just one of two

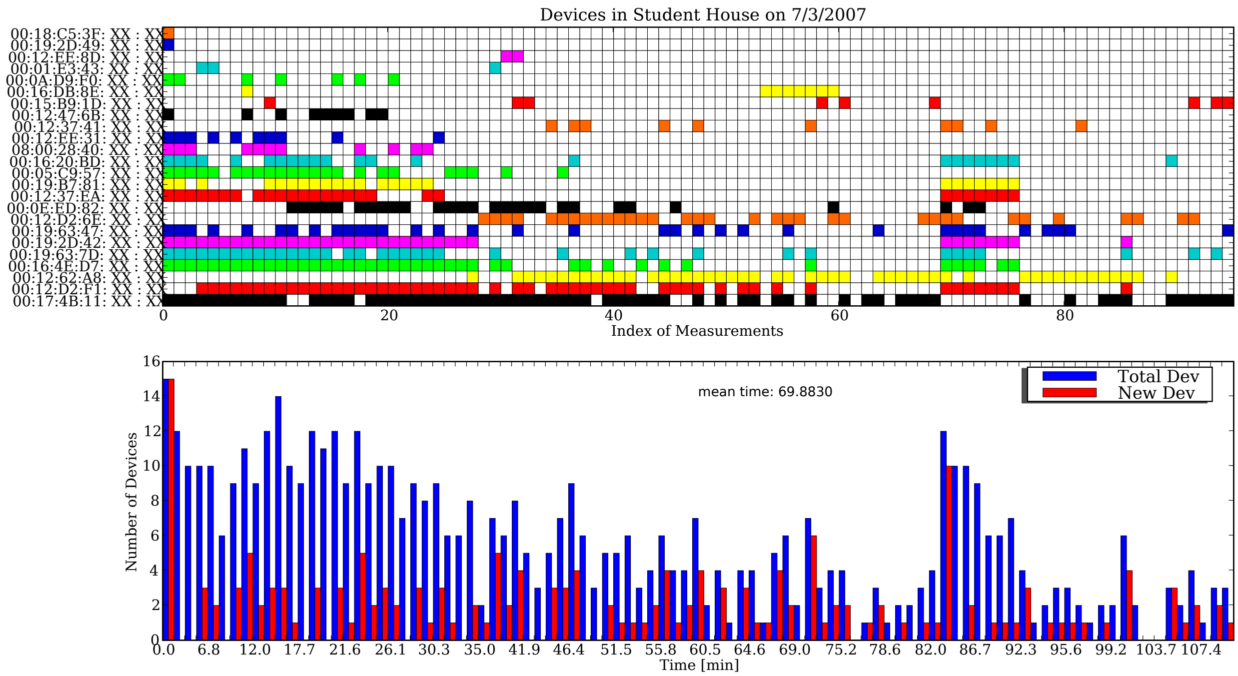

Figure 3.1: Devices in Student House

times, as can be seen from the first three rows from the top in Figure 3.1, for all the other cases some general considerations can be summarized as follows:

- Some devices seem always present, i.e. they stay around the sensing mobile for most of the time, so that it will always discover them. This is evident for the first device on the bottom, the one associated to the black color. It seems to be almost always in the range of the mobile that is taking the measurements. However, there are some white holes in the black line that can mean small movements of one of the two devices, the one that is discovering or the discovered one, so that one is not anymore in the BT range of the other;

- Some devices are present for quite long time, then they seems to disappear for a while, and then they come back again. This is particularly evident for the devices corresponding to rows 10, 11 and 14 from the bottom in Figure 3.1. They are present for the first measurements and then they appear again in the second half of the scanning. For these three devices it can be also said that, when they appear again, they are together, because they appear and disappear at the same time and for the same number of measurements;

- Some devices appear just for one interval of time, then they are not present anymore. An example is the green device in row 12, that is present just for most of the time regarding the first half of the measurement time, or the yellow one in the $19-th$ row, that, except for one presence at the beginning, is then present for an interval of 7 measurements in the middle.

The lower plot in Figure 3.1 shows for each search the number of total and new devices. Because of the small dimensions of the bar, people have no possibility to change position very often, so the new number of devices met is not high. In fact, for most of the searches it is less than three, while the number of total devices reaches 14 or 15. In this case, people can stay together for long time, so that more robust clusters can be established, even in terms of signalling. For most of the searches, in fact, more than three devices are together, as it's clear from the upper plot in Figure 3.1.

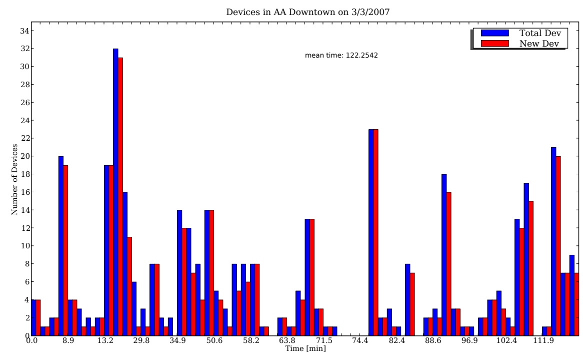

Figure 3.2: Devices in Downtown

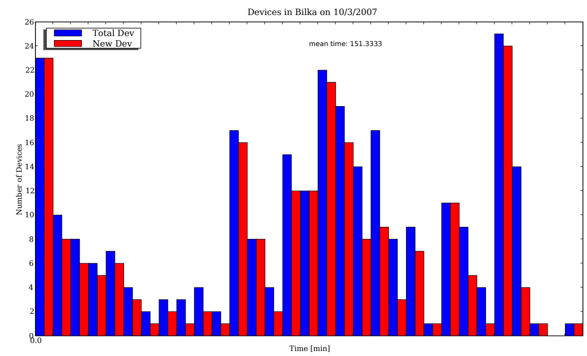

Figure 3.3: Devices in Bilka

The two cases presented in Figure 3.2 and Figure 3.3 regard the measurements taken during a walk in downtown streets of Aalborg city and in a shopping mall named Bilka respectively. In these cases the scanning is done while the mobile is carried by a moving person, so the number of new devices met is quite high for every new search. Case in Figure 3.2 shows big number of total devices found, with picks of 32 devices, and the new ones are around the same numbers of the old ones almost for each scan. In this scenario, the number of devices found has a great variance, going from cases in which large numbers are found to cases in which no devices are present. Big variance and standard deviation mean that most of the power will be spend to do signalling. In fact, in this case, because the number of devices is changing really often, it is more often necessary to check how to form a new cluster, as soon as to obtain information about the devices that have changed. In Figure 3.3 the total number of new devices reaches picks of 24-23. As in previous case, due to the possibility of mobility in a large area offered by the big mall, here also the number of new devices is almost equal to the one of total devices. These two cases also show a mean time of measurements higher than in the Student House, around 122 sec and 151 sec respectively. In fact, as we will prove in Section 3.6, the time of measurements becomes bigger as the number of devices discovered grows. As said before, 30 seconds is the interval from the end of a search and a new one. Then, each measurement takes some time, and this time depends on the number of devices found. Because in the places referred from figures 3.2 and 3.3 the number of total devices found is quite high, then the mean time is bigger.

In Table 3.1, the mean value of the number of total devices found , the value of the variance and the standard deviation for all the places taken into account is showed. The values are the mean of the numbers obtained for each day in each place.

Places	**Mean Tot**	**Var Tot**	**Std Tot**
Bilka	7.76	38.74	6.22
Gigantium	35.41	433.93	20.83
$AAU - Kantine$	2.22	1.64	1.28
Tlab	6.09	9.04	3.01
Station	3.11	8.13	2.85
Bus_n2	3.64	3.37	1.83
StudentsHouse	7.60	8.04	2.84
AADowntown	5.19	41.72	6.46
Metro	4.00	11.26	3.35
TrainBerlin	3.18	2.12	1.46
$MetroU2$	2.63	0.80	0.89
$MetroU9$	0.95	1.78	1.33
DublinAirport	28.1	184.09	13.57
BarDublin	9.24	17.83	4.22

Table 3.1: Places Statistics

Places like downtown, the shopping mall (Bilka), the conference center (Gigantium) and Dublin airport gives really big numbers as mean of devices found, even 28 or 35, as can be seen in Table 3.1. Anyway, these same places also offer the biggest variances. This means that the number of devices is not stable but it is varying a lot. On the other hand, variances are really small in places like metro, kantine, bus, when the same devices are together for long time, and the possibility of new devices is not big, so the total number is quite stable.

3.3 Types of Devices

From the analysis regarding the types of devices, it can be stated that the most common category is the phone. When we say phones we want to intend both common cellular phones and more modern smart phones, as well as cordless or other types of phones. Smart phones and cellular phones represent the most common types within the phone category, as clear from figures 3.4, 3.5, 3.6 and 3.7.

A smart phone is a full-featured mobile phone, with several functionalities. Smart phones are becoming more and more often used, because they represent a middle way between common mobile phones and Personal Digital Assistant (PDA) or computers. In this way they are the candidates to be used to achieve the modern concept of cooperation, which includes internet downloading, video streaming, etc.

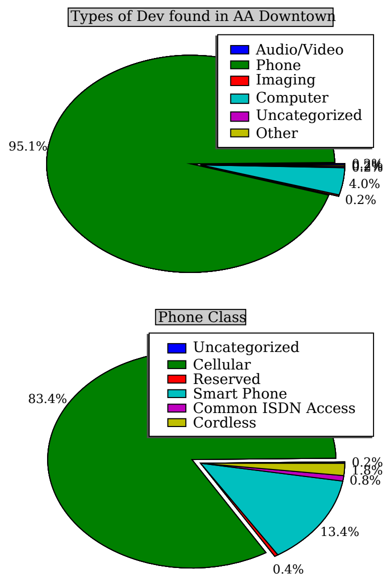

Figure 3.4: Types in Downtown

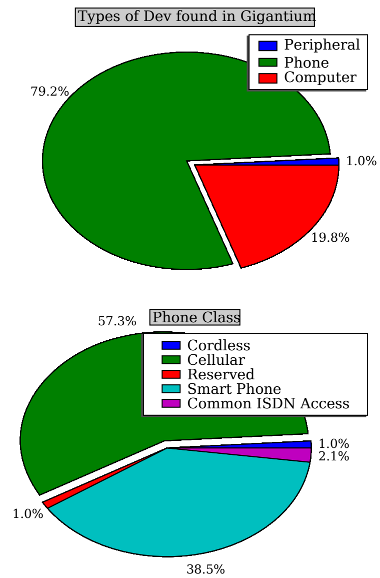

Figure 3.5: Types in Gigantium

As said before and as can be seen from the upper subplots in figures 3.4, 3.5, 3.6 and 3.7, the most common type of BT device found regards the phone category, especially in places like downtown, airport or bus. The reason is evident: almost everyone today has a mobile, while other types of devices are more rare. Especially in the bus, as clear from Figure 3.6, there are just phones found, in the percentage of 100% because usually people don't carry laptops switched on, or other types of BT devices like peripheral etc. An exception is Figure 3.5, in which a certain percentage of computers and peripheral is also discovered. This is because, as explained in Section 2.3, in this place computers and peripheral were present almost in each stand of the conference.

Within the phone category, common and smart phones cover the biggest percentages. However, common phones are much more present compared to smart phones. The reason is because the latter is a relatively new technology that is going to develop in these years. It can also be seen from Figure 3.5 that, in the conference center, the percentage of smart phones found is relatively high. Then, in the airport, a great variety of types of devices in the phone category can be seen. In fact, as clear from Figure 3.7, together with cellular and smart

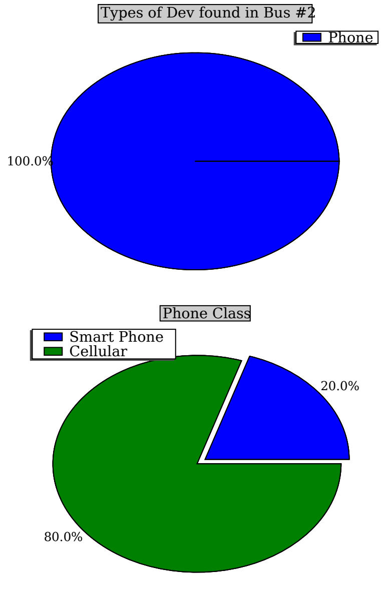

Figure 3.6: Types in Bus

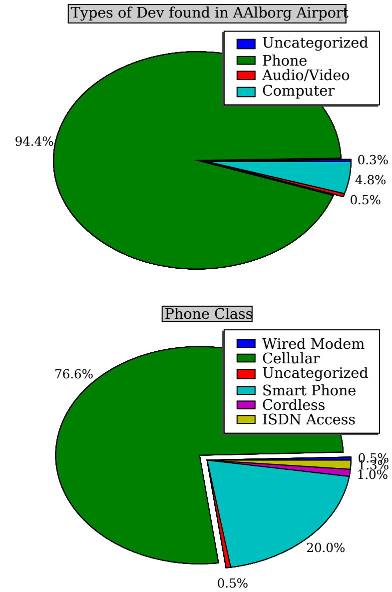

Figure 3.7: Types in AA Airport

phones, that are present in almost all the places under study as the biggest percentage, there are cordless, wired modems or voice gateway, common Integrated Services Digital Network (ISDN) access devices, and a small percentage of uncategorized ones. That cannot be possible in places like bus, etc. where cordless or wired modems are impossible to be found.

3.4 Probability Distributions Results

As said before, we are focusing on the idea of a cluster defined with three cooperative devices. Anyway, this is only an idea, because it can be stated that the only bad situation for cooperation is given by zero devices found, so cooperation is not possible. The case of only one device is also good, at least better than have no cooperation, and the case of devices more than three can be useful because a choice of the best to use in the cluster can be made.

The figures below contains three different subplots. The one with red bars shows the PDF performance, then the blue and the green ones show the CDF and ICDF respectively.

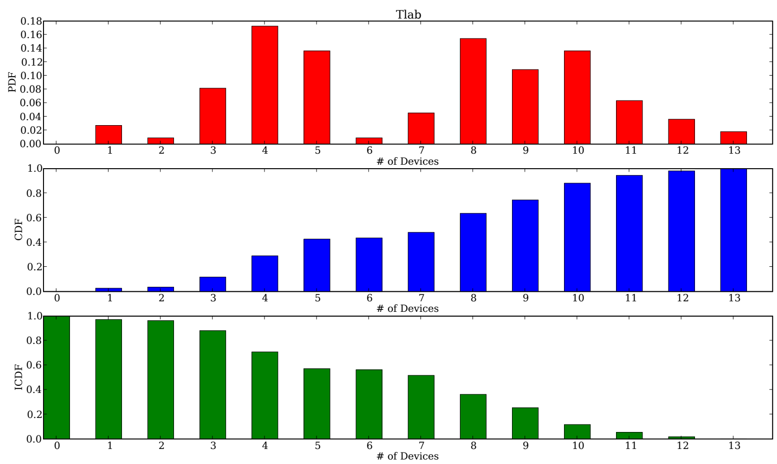

Figure 3.8: Probability Distributions-TLab

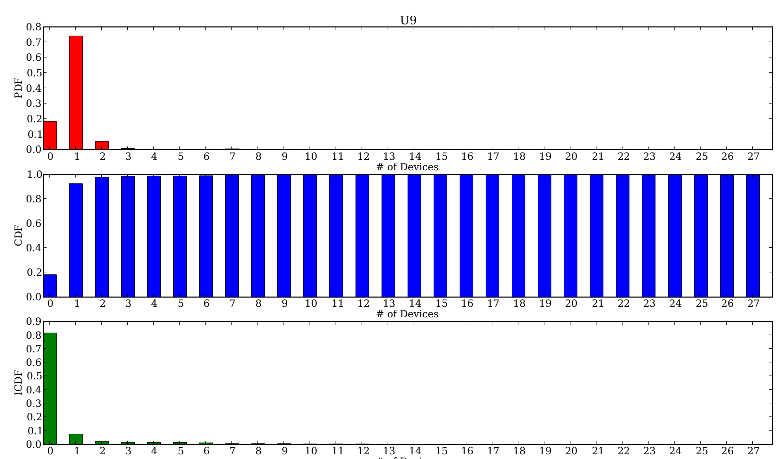

Figure 3.9: Probability Distributions-Metro U9

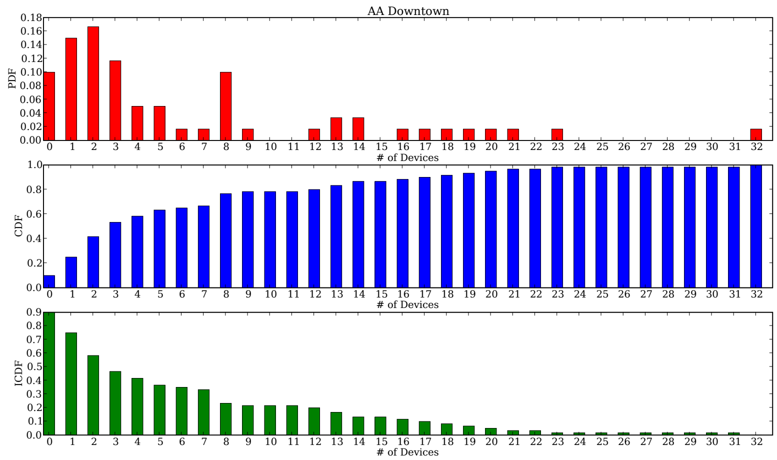

Figure 3.10: Probability Distributions-Downtown

In the upper plot of Figure 3.8 it can be seen that the probability to find for example 4, 5, 8 or even 10 devices is much more bigger. Thus, the cluster can be easily created. According to that, by the analysis of the CDF, or more easily the ICDF, it can be seen that the probability to find more than three devices is around 0.8, while the probability of more than zero devices is 1. In fact, if we look at the PDF subplot, there's no probability to find a number of devices equal to zero. All these considerations reveal this situation as a good one in order to benefit from cooperation.

On the other hand, in the upper plot of Figure 3.9, it can be seen that the possibility of having zero devices is more probable. Most of the probability is centered for number of devices equal to zero, one and two. For that reason, the CDF shows a rapid increasing shape, related with the rapid decreasing shape of the ICDF plot. In this case, probably cooperation will be done only with one devices.

Figure 3.10 shows a more continuous performance of both CDF and ICDF. The CDF grows slowly towards 1. This means that the probability to find a certain number of devices greater than one is not too small in respect to the case in which it grows suddenly.

3.5 Duration of Cooperation

In Section 2.5.5, three different definitions of cooperation time are given. Figures 3.11 and 3.12 show the percentage of time in which cooperation is possible respect to these definitions. As regards Case 2 and Case 3, because the initial choice of the cluster is random, the results are presented as the mean of the results obtained from ten different random choices.

As can be seen from Figure 3.11, Case 2 and Case 3 always give lower percentages than Case 1, because the latter just needs at least three devices to establish the cluster, it doesn't matter which ones are them, while in the other cases there is a constraint about the devices that have to form the cluster, as explained in Section 2.5.5. Case 1 gives a percentage of 100% of available time for cooperation because the number of devices found in Gigantium is always bigger than three for all the searches.

In Figure 3.12, however, the three different cases give almost equal percentage. The place

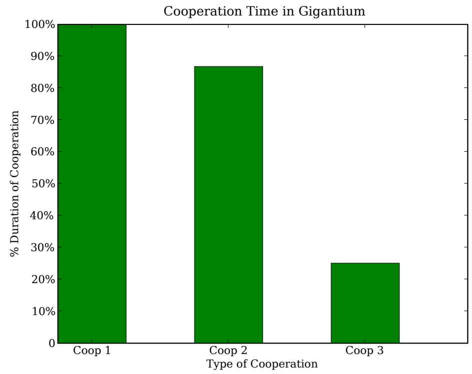

Figure 3.11: Coop Time Percentages in Gigantium

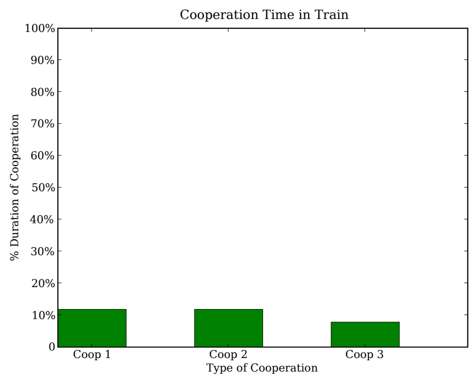

Figure 3.12: Coop Time Percentages in Train

in question is a train, and the number of devices found is not too big, so the three devices needed for Case 2 and Case 3 can be the only one existing, and the percentages are not high.

3.6 Scanning Duration vs Number of Devices

As said in Section 3.2, the scanning time is proportional to the number of terminal/devices discovered. Figure 3.2 shows the scanning time vs the number of devices found. The results are the mean values of the ones found for the different places analyzed. For each place, the total number of days in which the searches have been done are considered together, and the mean value is taken. For each number of possible devices found, the scanning time is calculated as the difference between the times in which the actual scanning and the next one start. Then, the mean time is taken in the case in which this number of devices is found for more than one search. Once this is done for all the places, the mean among all places is calculated.

Because the result is a time increasing with the number of devices, it can be stated that the scanning duration is proportional to the number of devices found. Thus, more devices are discovered, more time is needed for the SR link organization. This is one major result of this work underline the shortcomings of today short range communication technologies.

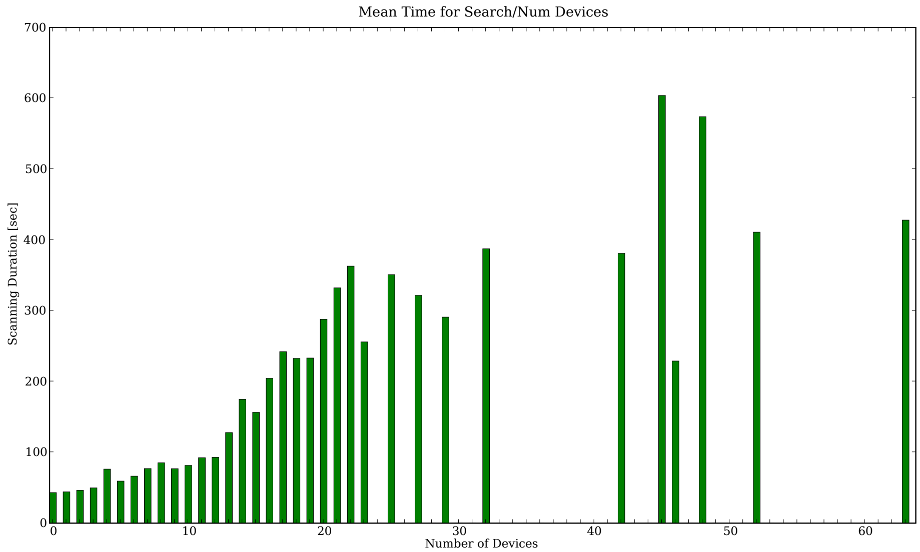

Figure 3.13: Scanning Time vs Number of Devices

3.7 A Particular Scenario: Aalborg Airport

This section interests a set of measurements taken during ten days in Aalborg Airport, as described in Section 2.3.

The measurement results can be read in Table 3.2. It summarizes the results by presenting, for each day, the mean value of the number of total and new devices found, the percentage of phones among all the devices and the percentage of smart phones within the phones category, as well as the percentage of time useful for cooperation. The latter is defined in the case that at least three devices are found in the surrounding area (T_{coop1} defined in Section 2.5.5), so that a cluster of at minimum three devices can be established.

Days	Mean Dev	Mean New Dev	%Phones	%Smart Phones	%T_Coop
1/4/2007	1.82	0.33	96.30	14.81	75.90
2/4/2007	4.37	0.74	91.81	23.27	60.40
3/4/2007	4.41	0.79	94.74	17.67	52.55
4/4/2007	3.79	0.63	95.51	17.14	52.70
5/4/2007	2.28	0.31	98.25	12.28	32.83
6/4/2007	2.99	0.58	94.97	21.48	39.16
7/4/2007	2.28	0.31	98.25	12.28	32.83
8/4/2007	1.88	0.38	91.35	21.63	33.76
9/4/2007	2.28	0.47	95.81	22.75	27.92
10/4/2007	4.27	0.76	95.70	20.31	54.11

Table 3.2: Airport Statistics

For reasons related to the position of the sensing mobile, as already described in Section 2.3, the mean number of devices discovered is not too high, as can be seen from Table 3.2. This is also because the phone was scanning for 19 hours everyday, including times in which there were no flight and almost nobody was in the airport. Most of the devices discovered belong to the category of phones, with a percentage always more than 90%. Among phones, smart phones cover a percentage only around $15 - 20\%$. The percentage of time in which more than two devices are found is for five days bigger then 50%. This means that the idea of clustering cooperating networks it's not so unreal, and can be usefully applied in such places, were people wait together for relatively long time.

The airport in question is not too big, so the only moment in which the waiting room is crowded it's some time before the departure time of flights, i.e. 10 times a day in mean. This can be seen from Figure 3.14. The upper plot represents the total number of devices found in each search; red lines are placed corresponding to the departures times of the flights related to the main air company in the place in question. In the lower plot, yellow rectangulars indicate the intervals in which people start to leave the waiting room of the airport because the gate is opened. If we look at the total number of devices in correspondence to the yellow intervals, we can see how the number becomes lower as the departure time is closer. The departure times usually corresponds to the lowest picks in the plot, while the biggest picks are placed around 45 minutes before that time, i.e. before the gate opens, and more people are supposed to crowd the room.

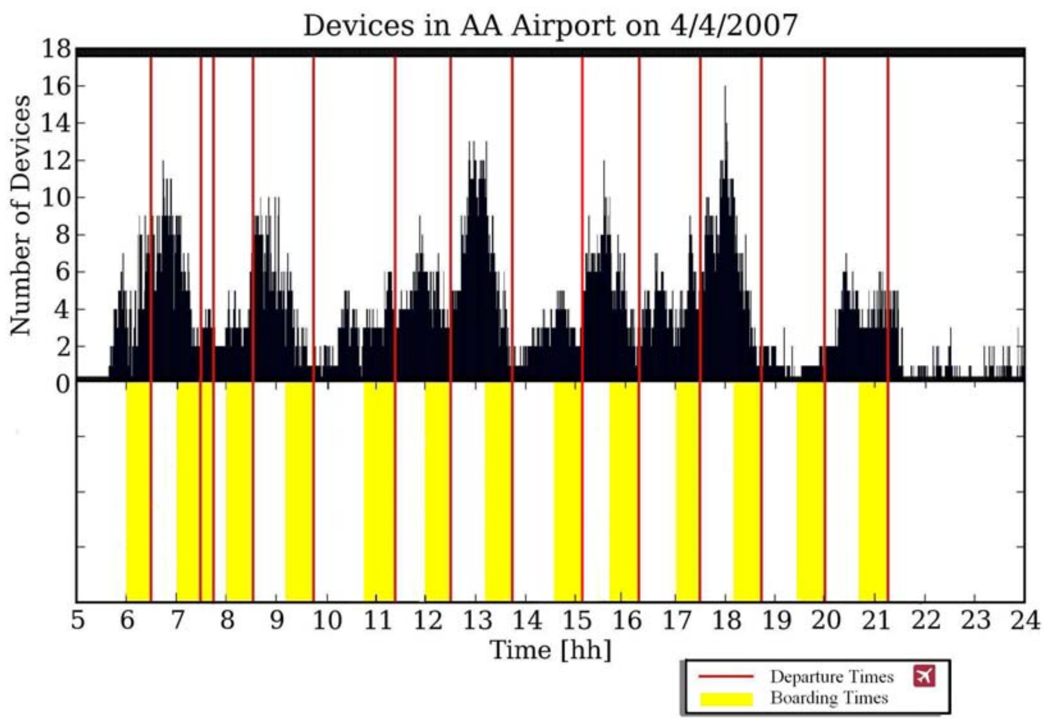

Figure 3.14: Total Devices Related to Departure and Boarding Times

Evident picks in the histogram correspond to the intervals before the times of departure of flights (yellow zones). In the upper plot, at the beginning of the x axis, the number of devices discovered is zero. That is the case of searches done really early in the morning (the script starts at 5 o'clock in the morning), when no flights are leaving. Then, there are some picks of 13 and even 16 devices found.

That is just one example regarding one day in the whole week in which the measurements at the airport were taken. The plots for the other days show a similar shape.

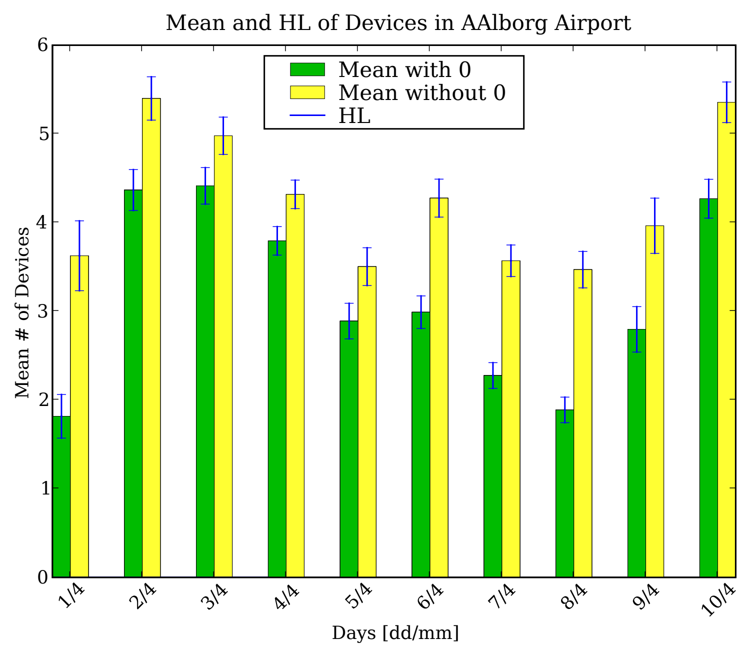

Figure 3.15: Means of Devices found in Airport

In Figure 3.15 the mean values of the number of devices found for each day (the 19 hours of scanning) at the airport is presented together with the corresponding confidence interval (HL). The histogram contains two types of bars. The green one (left one) is proportional to the mean value of devices found over the 19 hours, for all the scanning measurements done. The yellow bars (the right ones) represent the mean number of devices in case the sensing device has found at least one device. As there are time when no persons were allowed to enter the gate, we deleted those instances in which no devices were found. As can be seen from Figure 3.15, the yellow bars are by definition higher than the green ones. To summarize the mean value of mobile devices found is around four, but for certain situation up to 16 devices can be found, as in Figure 3.14. As regards the 95% HL, it can be stated that the smaller is the interval the better is the situation, because it means that the possible values are close to the mean obtained, with no hard fluctuations.

As said in Section 2.5.6 we are looking also into the scanning time needed to find a certain number of devices. Figure 3.16 shows the scanning time (in seconds) vs the number of devices found, with the relative HL. As already discussed in Section 3.6, the scanning time needed is proportional to the number of devices discovered. The HL is really small if the variance is small, i.e. the numbers found are really close to the mean values. This is true for small number of devices found (zero, one, two). The HL is zero for the last four bars. The reason is that these numbers were found just one time in the measurements, so the variance is zero. The bigger HL corresponds to 20 devices found, when the width of the interval is 45.13 seconds.

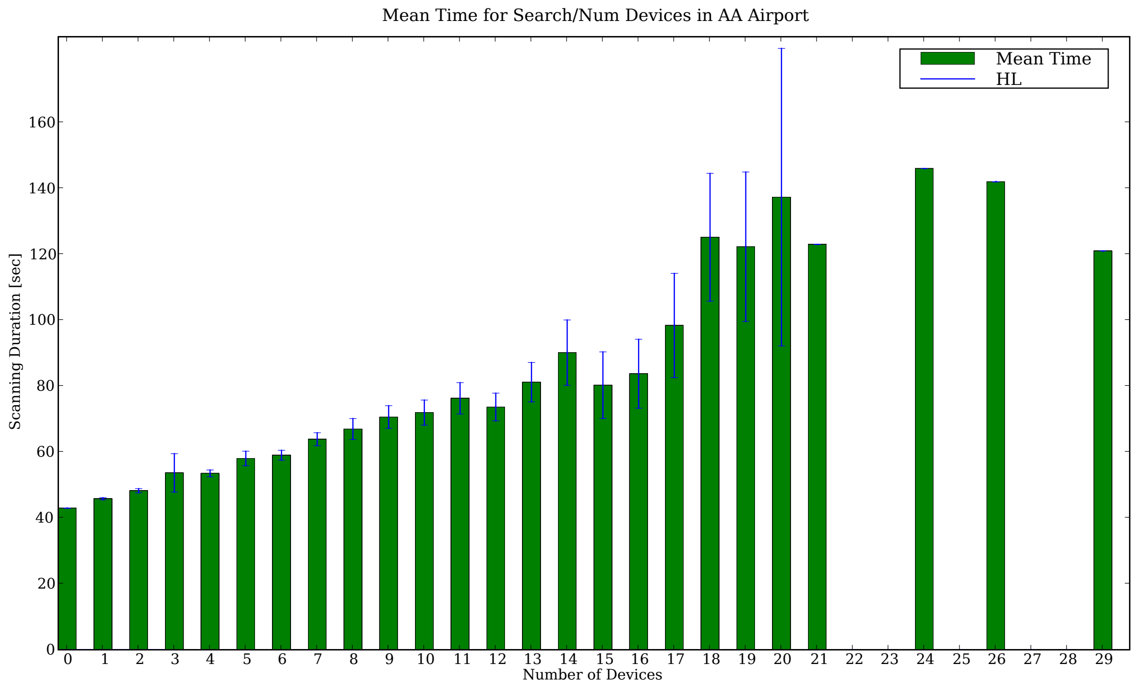

Figure 3.16: Scanning Time in Airport

This means that the relative mean time is not an reliable value.

Chapter 4

Conclusion and Future work

This chapter summarizes all the results, concludes the work and presents some ideas for future work.

4.1 Conclusions

In this report we have analyzed a measurement campaign dealing with the connectivity of BT devices. In particular, we focused on the availability of mobile BT devices in order to form mesh networks in typical wireless scenarios. Public places have been taken in considerations. Some examples are buses, airports, bars, shopping malls, etc.

Even if people tend to keep off their BT to avoid viruses, hackers attacks and to save energy, the results are quite promising. Places like downtown offers numbers of 32 devices found in one search.

The analysis of the type of BT devices is also presented. Phones represents the biggest category, with percentages around 90-95% in all the scenarios under study. Within the phone category, common phones represent the greatest percentage, followed by smart phones.

Considerations about the signalling needed to establish a cluster among cooperating devices have been presented. Big signalling is related with highly dynamic places, in which nodes tend to rapidly change their distribution while moving. In this case the sensing mobile was discovering a large amount of new devices in each search. As regards the cluster creation, possible metrics are given. For each metric the interval of time in which the cluster can be kept alive was calculated and defined as time of connection of nodes.

A particular regard was given to the relation scanning time-number of devices. The proportionality between the number of devices and the time needed to scan them gives the result of more time needed for the SR link organization as more devices are discovered. In general, BT needs long time to discover the devices around. The scanning was also limited in speed because of cache refreshing problems and battery consumption in the mobile, so a pause of 30 seconds was needed.

4.2 Future work

Due to technical problems, some assumptions have been made in this project, including:

- Each scan is done after a pause of 30 seconds from the previous one;

- The BT range of the sensing device was limited to around 10 meters;
- The measurement campaign was confined to public places almost all in one single city, i.e. Aalborg (Denmark);
- Data were collected from a relative small sample of people.

As future work, it will be interesting to make the same measurements and evaluations as done in the project when one or more of the previous assumptions are not satisfied.

First of all, it could be interesting to refresh the sensing phone cache every search, so that the pause interval is not needed anymore. Thus, it is possible to have constantly information about the BT devices around, without missing any period of time. To avoid any lose of time, because we found that BT is not a really fast discovering technique, another discovering technology could be implemented.

Then, a very interesting topic could be the analysis of measurements taken from several users in different areas of the world. As said before, our application is available to download [2], and we also implemented a client-server application to send data to a server, update and store automatically new data in our web page. Thus, everybody could download the application, make measurements, and make new data available on Internet.

Appendix A

XML

A.1 XML Introduction

XML stands for Extensible Markup Language. It is designed to describe data and to focus on what data is [21].

XML has been built up to create, store and send information with a structure made up of symbols embedded in the text, called markup [22]. Markup organizes information in a particular way by using symbols. Therefore, XML can be seen as a markup language that supports a wide variety of applications in order to facilitate the sharing of data across different information systems and this is especially needed for electronic documents in informatics.

XML divides information into container-like elements and attributes to them, and they are organized as a tree-based structure. Elements can contain other elements, which also can contain other elements and so on, as to create a hierarchically organization, such as a tree-structure.

A.1.1 Versions

There are two different versions of XML. The first is XML 1.0. It was first published in 1998, but has been developed in several versions and actually a fourth version is available. The second is XML 1.1, that was defined in 2004, and a first and a second version are available. Even if in certain cases that last version is easily to use, it is not really recommended because it is not widely implemented.

A.2 Structure

The typical structure of an XML document includes one outermost element, also called document element or root element, that contains all the other elements and contents in the document.

There can exist some optional information on the top, that create the document prolog. This is a special section placed before the root element that contains metadata, such as the XML declaration and the document type declaration. The first one, when present, should be the first line of the document, and it specifies the version of XML used and the character encoding used in the document. It is also possible to put a standalone parameter, that informs the parser if there are declarations outside the document. The second is the document type definition (DTD), which begins with the delimiter <!DOCTYPE and ends with the delimiter > [22]. Between them, there is an element name that specifies the type of the document element, and the next is an optional identifier for the document type definition, such as a file on the system, or a URL to a file on Internet, etc. Then, the last part between brackets is an optional part in which some optional entities can be declared. It is possible to refer to these entities later in the text by using an ampersand (&), the entity name, and a semicolon (;).

After the declaration, the document contains one or more elements, each of them containing text, other elements or nothing (empty elements). If it contains both text and element, its content is called a mixed content. Each element has to follow a particular syntax. The basic syntax for one element in XML is:

<name attribute=”value”>content</name>

So, an element typically consists of two tags: a start tag and an end one, that surround text and other elements. The start tag consists of a name in angular brackets, and may also some attributes. The end tag contains between angular brackets the same name as the respective start tag, but just after the opening angular brackets there is a slash (/). Tags are not predefined. The name that is in the start tag must be identical to the one in the end one, considering also that the XML language is case-sensitive. An empty element has no text, so it does not need a closing tag, but only an opening one. For this reason, it consists only of two brackets and between them there is the name and possible attributes with a slash (/) before the closing bracket. A name can contain all the alphanumeric characters and accented characters, but the only punctuation allowed are the dot(.), the under-score(_) and the hyphen(-).

Names can only start with a letter, ideograph or underscore, and, as said before, the XML language is case-sensitive. It is not allowed to put spaces, tubs or newline between the opening bracket and the first character of the name. Anyway, these characters can be used in the tag, for example to separate attributes.

The end tag has to be put after and never before the corresponding opening one, and they have both to be within the same parent element, otherwise there will be an overlapping error.

In addition to text content, an element can contain attributes. An attribute is a name-value pair included in the start tag after the element name. Attributes are useful in order to distinguish among elements of the same name instead of creating a new element for each of them. There is no theoretical limit to the length of a name and to the number of attributes, but each attribute name should appear only once in any element.

Every XML document must have one and only one root element.

There are some characters that cannot be used in a XML document text, because they are special markup delimiters. Thus, whenever it is necessary to use these characters, they need to be referred as predefined characters entities. It is also possible to use a numeric character entity to represent a character that is not easily encodable, so this character is referred by its number in the Unicode set. The number in the entity name can be represented either in decimal or in hexadecimal format. Entities can include more than the value of a single character, but they can include text as well as markup characters, so they are named as mixed-content references. They can be internal or external entities. The first ones are defined in the document type declaration, in the entity declaration part. For the second ones, the replacement text is in another file. They are really useful because any eventual change in that text will be done only once in the text of the entity, and not every time it is used, even if it will appear everywhere.

Unparsed entities can be used in order to import graphics, sound files or noncharacted data.

Anyway, the XML document should be written and organized in a way so that there's only a possible interpretation of the elements, with their names and possible attributes.

A.3 Why to Use XML

XML is a very easy and useful way to store data. However, the way in which an XML document is written is not really easy for human people to read. Rather, an XML file can be easily processed to organize data in a suitable form.

XML also supports several scripts and several encoding types. XML is a data description language, so its documents give no information to how to display data. A generic XML document can be opened with a web browser. Other languages (such as XSL, CSS, etc.) can be used to format the XML data.

An XML parser is needed to manipulate an XML document. The parser loads the document into the computer memory and, once the document is loaded, its data can be manipulated using the Document Object Model (DOM), that treats the XML document as a tree. There are some differences between Microsoft's XML parser and the XML parser used in Mozilla browser [22].

Appendix B

BT services

B.1 CoD

The function finddevices() is used in the client program to discover the BT devices in the range of the sensing device. It gives a list of tuples, such as [(addr,name,CoD),(..),(...),...], where (...) is a tuple [14]. Each tuple contains the address (addr), the name (name) and the CoD of the device found. The CoD has a different format depending to the "Format Type Value". For the 'format # 1', the "Format Type Value" is equal to 00 (last to bit of the CoD). In that case, the CoD is a number of 24 binary digits, that can be interpreted as the major service class (bits from 13 to 23), the major device class (bits from 8 to 12) and the minor device class (bits from 2 to 7).

B.1.1 Major Service Class

The major service class is related to 10 bits (from 13 to 23 in the CoD) whose meaning can be seen from Figure B.1 and table reftab:MaSC. Any reserved but unassigned bit should be set to zero [14].

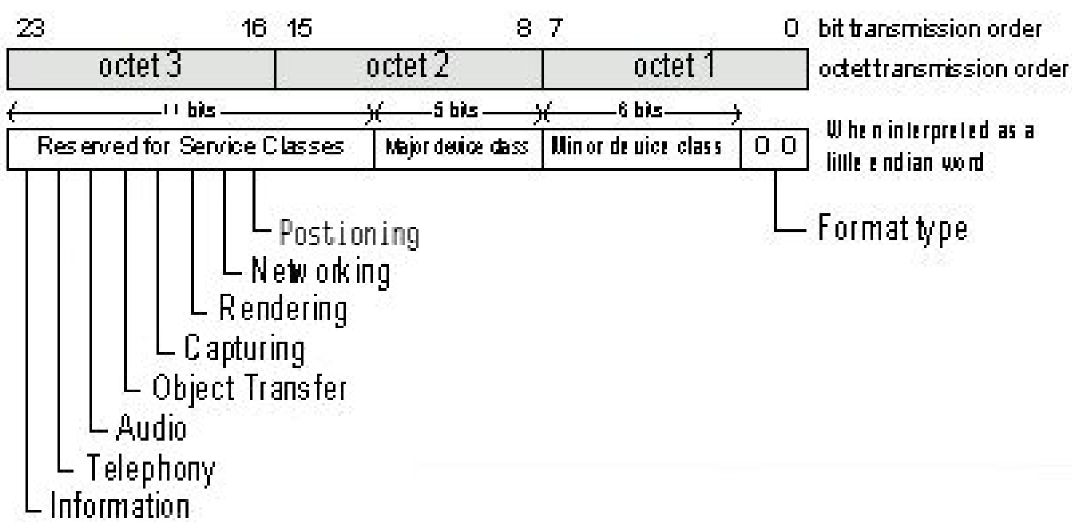

Figure B.1: Major Service Class

Bit no	Major Service Class
13	Limited Discoverable Mode [Ref# 1]
14	(reserved)
15	(reserved)
16	Positioning (Location identification)
17	Networking (LAN, Ad hoc, ...)
18	Rendering (Printing, Speaker, ...)
19	Capturing (Scanner, Microphone, ...)
20	Object Transfer (v-Inbox, v-Folder, ...)
21	Audio (Speaker, Microphone, Headset service, ...)
22	Telephony (Cordless telephony, Modem, Headset service,...)
23	Information (WEB-server, WAP-server, ...)

Table B.1: Major Service Classes

B.1.2 Major Device Classes

The major device class defines a general family of BT devices by using five bits (from 8 to 12 in the CoD), so thirty-two different possible major classes can be defined. Once found the major class for a particular device, bits from 2 to 7 specify the minor class. They are six bits, so sixty-four possible minor classes can be defined. The minor class can be defined only within its corresponding major class, but it's independent to the service class.

Bit no of CoD					**Major Device Class**
12	11	10	9	8	
0	0	0	0	0	Miscellaneous [Ref #2]
0	0	0	0	1	Computer (desktop,notebook, PDA, organizers, ...)
0	0	0	1	0	Phone (cellular, cordless, payphone, modem, ...)
0	0	0	1	1	LAN /Network Access point
0	0	1	0	0	Audio/Video (headset,speaker,stereo, video display, vcr.....
0	0	1	0	1	Peripheral (mouse, joystick, keyboards, ...)
0	0	1	1	0	Imaging (printing, scanner, camera, display, ...)
0	0	1	1	1	Wearable
0	1	0	0	0	Toy
1	1	1	1	1	Uncategorized, specific device code not specified
X	X	X	X	X	All other values reserved

Table B.2: Major Device Classes

B.1.3 Minor Device Classes

Minor Device Class field - Computer Major Class

Bit no of CoD						**Minor Device Class**
7	6	5	4	3	2	
0	0	0	0	0	0	Uncategorized, code for device not assigned
0	0	0	0	0	1	Desktop workstation
0	0	0	0	1	0	Server-class computer
0	0	0	0	1	1	Laptop
0	0	0	1	0	0	Handheld PC/PDA (clam shell)
0	0	0	1	0	1	Palm sized PC/PDA
0	0	0	1	1	0	Wearable computer (Watch sized)
X	X	X	X	X	X	All other values reserved

Table B.3: Sub Device Class field for the 'Computer' Major Class

Minor Device Class field - Phone Major Class

Bit no of CoD						**Minor Device Class**
7	6	5	4	3	2	
0	0	0	0	0	0	Uncategorized, code for device not assigned
0	0	0	0	0	1	Cellular
0	0	0	0	1	0	Cordless
0	0	0	0	1	1	Smart phone
0	0	0	1	0	0	Wired modem or voice gateway
0	0	0	1	0	1	Common ISDN Access
X	X	X	X	X	X	All other values reserved

Table B.4: Sub Device Classes for the 'Phone' Major Class

Minor Device Class field - LAN/Network Access Point Major Class

Bit no of CoD			Minor Device Class
7	6	5	
0	0	0	Fully available
0	0	1	1 - 17% utilized
0	1	0	17 - 33% utilized
0	1	1	33 - 50% utilized
1	0	0	50 - 67% utilized
1	0	1	67 - 83% utilized
1	1	0	83 - 99% utilized
1	1	1	No service available
X	X	X	All other values reserved

Table B.5: The LAN/Network Access Point Load Factor field

Bit no of CoD			Minor Device Class field - LAN/Network Access Point Major Class
4	3	2	
0	0	0	Uncategorized (use this value if no other apply)
X	X	X	All other values reserved

Table B.6: Reserved sub-field for the LAN/Network Access Point

Minor Device Class field - Audio/Video Major Class

Bit no of CoD						Minor Device Class
7	6	5	4	3	2	
0	0	0	0	0	0	Uncategorized, code not assigned
0	0	0	0	0	1	Wearable Headset Device
0	0	0	0	1	0	Hands-free Device
0	0	0	0	1	1	(Reserved)
0	0	0	1	0	0	Microphone
0	0	0	1	0	1	Loudspeaker
0	0	0	1	1	0	Headphones
0	0	0	1	1	1	Portable Audio
0	0	1	0	0	0	Car audio
0	0	1	0	0	1	Set-top box
0	0	1	0	1	0	HiFi Audio Device
0	0	1	0	1	1	VCR
0	0	1	1	0	0	Video Camera
0	0	1	1	0	1	Camcorder
0	0	1	1	1	0	Video Monitor
0	0	1	1	1	1	Video Display and Loudspeaker
0	1	0	0	0	0	Video Conferencing
0	1	0	0	0	1	(Reserved)
0	1	0	0	1	0	Gaming/Toy
X	X	X	X	X	X	All other values reserved

Table B.7: Sub Device Classes for the 'Audio/Video' Major Class

Minor Device Class field - Peripheral Major Class

Bit no of CoD		Minor Device Class
7	6	
0	0	Not Keyboard / Not Pointing Device
0	1	Keyboard
1	0	Pointing device
1	1	Combo keyboard/pointing device

Table B.8: The Peripheral Major Class keyboard/pointing device field

Bits 6 and 7 independently specify mouse,keyboard or combo mouse/keyboard devices. These may be combined with the lower bits in a multifunctional device [14].

Bit no of CoD				Minor Device Class
5	4	3	2	
0	0	0	0	Uncategorized device
0	0	0	1	Joystick
0	0	1	0	Gamepad
0	0	1	1	Remote control
0	1	0	0	Sensing device
0	1	0	1	Digitizer tablet
0	1	1	0	Card Reader (e.g. SIM Card Reader)
X	X	X	X	All other values reserved

Table B.9: Reserved sub-field for the device type

Minor Device Class field - Imaging Major Class

Bit no of CoD				Minor Device Class
7	6	5	4	
X	*X*	*X*	0	Display
X	*X*	1	*X*	Camera
X	1	*X*	*X*	Scanner
1	*X*	*X*	*X*	Scanner
X	X	X	X	All other values reserved

Table B.10: Reserved sub-field for the LAN/Network Access Point

Bits 4 to 7 independently specify display, camera, scanner or printer. These may be combined in a multifunctional device [14]. Bits 2 and 3 are reserved.

Bit no of CoD		Minor Device Class
3	2	
0	0	Uncategorized, default (use this value if no other apply)
X	X	All other values reserved

Table B.11: The Imaging Major Class bits 2 and 3

Minor Device Class field - Wearable Major Class

Bit no of CoD						Minor Device Class
7	6	5	4	3	2	
0	0	0	0	0	1	Wrist Watch
0	0	0	0	1	0	Pager
0	0	0	0	1	1	Jacket
0	0	0	1	0	0	Helmet
0	0	0	1	0	1	Glasses
X	X	X	X	X	X	All other values reserved

Table B.12: Sub Device Classes for the 'Wearable' Major Class

Minor Device Class field - Toy Major Class

Bit no of CoD						Minor Device Class
7	6	5	4	3	2	
0	0	0	0	0	1	Robot
0	0	0	0	1	0	Vehicle
0	0	0	0	1	1	Doll / Action Figure
0	0	0	1	0	0	Controller
0	0	0	1	0	1	Game
X	X	X	X	X	X	All other values reserved

Table B.13: Sub Device Classes for the 'Toy' Major Class

Bibliography

[1] F.H.P. Fitzek and M. Katz. *Cooperation in Wireless Networks: Principles and Applications - Real Egoistic Behavior is to Cooperate!* ISBN 1-4020-4710-X. Springer, second edition, April 2006.

[2] `http://kom.aau.dk/group/07gr1112/`.

[3] A. Oram. *Peer-to-Peer:Harnessing of a Distruptive Technology.* O Reilly Associates, first edition, 2001.

[4] R. Morrow. *Bluetooth Operation and Use.* McGraw-Hill, first edition, 2002.

[5] `http://news.bbc.co.uk/2/hi/technology/6544919.stm`.

[6] M. Kaminsky H.Pucha, D.G. Andersen. Exploiting similarity for multi-source downloads using file handprints. *In Proceedings of the 4th Symposium on Networked Systems Design and Implementation*, April 2007.

[7] `http://europe.nokia.com`, 2004.

[8] `http://www.nokia.com`, 2004.

[9] F.H.P. Fitzek and M. Katz. *Cognitive Wireless Networks: Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications.* ISBN 978-1-4020-5978-0. Springer, July 2007.

[10] N. Nasser L.M. Arboleda. Comparision of clustering algorithms and protocols for wireless sensor networks. *IEEE CCECE/CCGEI*, 2006.

[11] R. Palit. Mobility-aware access-based clustering in mobile ad hoc networks. Master's thesis, University of Manitoba, 2004.

[12] A. Chan D. Wei. A survey on cluster schemes an ad hoc wireless networks. IEEE Mobility Conference, 2005.

[13] C.F. Sturman J. Bray. *BLEUTOOTH Connect Without Cables.* Prentice Hall PTR, first edition, 2001.

[14] `http://www.bluetooth.org`, March 2007.

[15] M. Lutz. *Programming Python.* O'Reilly, first edition, 1996.

[16] `http://www.python.org`, March 2007.

[17] W.J. Chun. *Core Python programming*. ISBN 0-13-026036-3. Prentice Hall, first edition, 2001.

[18] S. Holden. *Python Web Programming*. New Riders, first edition, 2002.

[19] J. Proakis. *Digital Communications*. John wiley and sons ltd., second edition, 2000.

[20] `http://matplotlib.sourceforge.net`, 2004.

[21] `http://www.xml.org`, March 2007.

[22] E. T. Ray. *Learning XML*. O'Reilly, second edition, 2003.