Analysis of WiFi Traffic Traces for D2D Support

Author: Enric Garcia Aldao

Supervisors: prof Carla-Fabiana Chiasserini
            prof Marco Mellia

July 2015
Abstract

Analysis of WiFi Traffic Traces for D2D Support

by Enric Garcia Aldao

The explosive growth of the smartphone users is causing a massive increase in traffic in the cellular network. Recent studies have proposed Device-to-Device (D2D) communications as a key feature of the next-generation mobile networks to cope with this problem. The aim of this thesis is to analyse traffic traces representing the content items that are requested by mobile users to investigate the capability of D2D to offload the net in a real scenario. This study was conducted at the Politecnico di Torino. This thesis first examines the correlation of the requested content between different users, characterizing such items in terms of popularity, size, type and cacheability. In second stage, it has been studied the time correlation on different days, proving the reliability of the mechanism. Finally, the spatial correlation has been examined, taking in count the distribution of the users through the different APs, exploring the capacities of the mechanism in a more realistic scenario. On the basis of the results of this research, it can be concluded that D2D mechanism is a promising solution to offload the cellular network, despite remaining technical issues and uncertain business strategies that have to be further investigated.
Acknowledgements

A la meva família per permetre’m aprofitar totes les oportunitats que he tingut fins ara.
To my supervisors, Carla-Fabiana Chiasserini and Marco Mellia, for guiding me in this work and give me all their support and understanding.
# Contents

Abstract ii
Acknowledgements iii
Contents iv

1 Introduction 1

2 D2D Data Offloading 4
  2.1 Trends in D2D standardization 5
  2.2 Device-to-Device communications 6
    2.2.1 Benefits, usage cases and business models 7
    Peer to Peer usage cases 8
    Local Voice server: 8
    Local Data service: 8
    Relay usage cases 8
    UE as the Gateway to Sensor Networks: 8
    UE cooperative Relay: 9
    Business Models 9
  2.2.2 Device-to-Device Architecture 10
    Fully controlled D2D Mode 10
    Loosely controlled D2D Mode 11
  2.2.3 Device-to-Device communication procedure 12
    Device Discovery 12
    Centralized approach: 12
    Distributed approach: 13
    Link Setup 13
  2.2.4 Related Work 13

3 Traffic analysis 15
  3.1 Data Collection 15
  3.2 HTTP logs 16
  3.3 APs logs 18
1 - Introduction

In recent years, mobile data traffic has grown explosively due to the increasing popularity of smartphones and mobile devices. At the same time, the data volume per device has increased rapidly. Recently, analyst from Cisco warned that global mobile data traffic is expected to grow 18-fold between 2011 and 2018, three times faster than the overall fixed IP traffic in the same period. In fact, in 2014 the number of mobile-connected devices such as smartphones and tablets had exceeded world population [1]. Unfortunately, this explosive surge in mobile data traffic has caused a sever traffic overload problem in the cellular network, where the bottlenecks are different from traditional wireline networks. The “last mile” of the cellular network is subject to many constraints that limit the system. Scarce licensed spectrum hinders the RAN enhancements. Regulations allow mobile operators to use only a small portion of the overall radio spectrum, which is also extremely expensive. Users must share the same limited wireless resources. Adding traffic beyond a certain limit mines the performance and the quality of service (QoS) perceived by the users. During the peak times in crowded metropolitan environments, users are already experiencing long latencies, low throughput, and network outages due to congestion and overload at RAN level [2].

The most straightforward solution is to enhance the capacity of cellular networks, but this is expensive and inefficient since this approach requires more infrastructure equipment and thus more investment. Therefore, 3GPP has defined data offloading as an alternate solution to cope with the problem. We can see from the Figure 1.1 the main approaches to offload in cellular networks when compared to the traditional infrastructure-only mode Figure 1.1(a). The most popular offloading technique consist in using WiFi APs instead of base station Figure 1.1(b) , implying that if some content has been previously downloaded by a user connected to the same APs, the APs will provide the content instead of the base station. The increasing popularity of smart mobile devices makes it possible to deploy a device-to-device (D2D) network that relies on direct communication between mobile
users, without any need for an infrastructure backbone 1.1(c). Enable the device-to-device communications as an underlay to the cellular network to increase the spectral efficiency. Benefiting from a shared interest among collocated users, a cellular provider may decide to send popular content only to a small subset of users via the cellular network, and let these users spread the information through D2D communications. The most widely known D2D technologies are Bluetooth and WiFi working at the 2.4GHz unlicensed band. Up to now, wireless operators don’t include the D2D function in the universal cellular network standards. But this technology could be far more useful, many of the new trend services depend on the user location. For example, a user may be informed of a nearby restaurant, and could book a table by sending a message. It could also be useful to implement a machine-to-machine (M2M) system. Since most consumer devices work around their owners, the cellular phone can be the hub for these devices and used as the gateway to the cellular networks, e.g. washing machines, central heating and ovens. Due to this all new applications, wireless operator’s attitude towards the D2D function is changing [3].

The aim of this thesis consists in analysing traffic traces representing the content items that are requested by mobile users in different scenarios in the Politecnico di Torino to decide if D2D offloading model can be an effective offloading solution. By looking and characterizing the content downloaded by different users in different locations at different periods, sharing or not same cells; we can try to estimate the usefulness of the system. We will perform two different analysis based on time and location. In the time correlation analysis, we will try to see if the content downloaded is constant through time, studying whether users download the same content on two different days, or on two consecutive intervals. This information will be very important to see the behaviour of the D2D offloading technic. The other approach is the spatial correlation, as two users must be
close to share content, it is important to study the behaviour of users constrained by space. It is not the same if you are connected in the cafeteria where you share your APs with many other users; increasing the probability to ask for content previously downloaded by another user, than if you are connected in a classroom where the APs and the content are only shared with your classmates. As it will be explained in the next chapters there are many constraints that will limit our study. The traces used in this thesis are captured by a tool called Tstat, which extracts information from the observation of the packets travelling through the edge router. This deployment makes impossible to distinguish if a packet is from a mobile device or from a computer, in a best case scenario we should study just those packets coming and going from a mobile device. For the purpose of the thesis, we conclude that nowadays there is no such difference in the content download through different devices. Another constraint is that we are basing our study in information given by the provider server, and there are many cases where this information could be misleading. Some assumptions in size have been done, for example, we have excluded those objects with no size. A basic filter has been developed to differentiate dynamic and static content. Assumptions have been made regarding the area and coverage of APs. All these constraints have limited our study, however, all the assumptions made in the thesis have been contrasted to carry out the study as faithful as possible to reality.
2 - D2D Data Offloading

In the past two decades, with the tremendous development of technology, mobile-devices are becoming widespread, this should not come as a surprise because nowadays it is impossible for anyone to imagine going out without a mobile-device with her. In 2011 the mobile industry shipped more smartphones and tables than PC’s [4]. This fact, coupled with the exponential increase of the data volume per device due the success stories of social networking services, streaming video services and data and voice services has overcome the capacity of the classic 2G/3G mobile networks. This concept is called “Internet offload”. In this forthcoming era where all the applications are pointing towards “always connected” scenarios, Device-to-Device communications promise to be the key feature of next-generation mobile networks. Before further analysis, it is important to highlight that mobile operators identify two types of Internet offload: radio access network (RAN) offload and core network offload. These are distinct in that they target different parts of the network and impact the users differently. In the cellular networks, the latest innovations in optical fiber technology provide the telecommunications industry with high–capacity, technology-robust and low-latency solutions for backbone, while also reducing costs to ensure carrier profitability. Due to the increasing popularity of the wireless local area networks, as they enable fast access to the internet and local service with low-cost infrastructure, the bottleneck has moved from traditional wireline networks to the wireless segment. To solve this problem, most efforts have been focused on what is called, the “last mile”, which refers to the part of the network that is closer to the user’s device. It is essentially the wireless link between the user’s device and the cell-tower. Many offload technics focus on this part of the network, because a bottleneck at an early stage will lead to increasing performance degradation in time. The first and most known data offloading technique is to transfer data from mobile networks to WiFi networks [5]. WiFi or Wi-Max networks are very fast and require no spectrum fees to implement them. The increase of velocity, the saving on battery and service charges are more than enough
reasons to switch our mobiles to a WiFi network. Nowadays it is so popular than new devices can switch automatically between networks without the interaction of the user.

2.1 Trends in D2D standardization

To satisfy future service requirements, network shall provide high bit data rates, and increase the system capacity, that’s why major effort has been put in recent years on the development of Third Generation Partnership Project (3GPP) Long Term Evolution (LTE). Through the 3GPP standards process, the first data offloading proposals formalized for use in the mobile operator environment can be summarized in the following Figure [6]:

<table>
<thead>
<tr>
<th>INITIATIVE</th>
<th>3GPP TS</th>
<th>RELEASE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-WLAN</td>
<td>TS 22.235</td>
<td>Rel-6</td>
<td>Integrate WiFi Access to mobile packet core</td>
</tr>
<tr>
<td>UMA/GAN</td>
<td>TS 43.318</td>
<td>Rel-6</td>
<td>Run CS and PS services over WiFi Access</td>
</tr>
<tr>
<td>Direct Tunnel</td>
<td>TS 23.919</td>
<td>Rel-7</td>
<td>SGSN bypass for user-plane traffic</td>
</tr>
<tr>
<td>ANDSF</td>
<td>TS 24.312</td>
<td>Rel-8</td>
<td>Core network function to ensure device selects best available WiFi</td>
</tr>
<tr>
<td>Femtocell</td>
<td>TS 22.220</td>
<td>Rel-9</td>
<td>Residential small cell operating in licensed spectrum</td>
</tr>
<tr>
<td>LIPA &amp; SIPTO</td>
<td>TS 23.829</td>
<td>Rel-10</td>
<td>IP offload specifications at BTS and the packet core</td>
</tr>
<tr>
<td>IFOM</td>
<td>23.261</td>
<td>Rel-10</td>
<td>Ability to split flows across access networks (e.g. WiFi &amp; cellular) on the same device</td>
</tr>
</tbody>
</table>

In the Rel-9 was introduced a femto base station solution. For femto, small isolated cells can be set up by the consumers themselves, when allowed by the operator, to increase spatial reuse of the operators’ spectrum. Femtocells require access to the cellular core network, and the radio resources assigned to the femtocell have to remain stable and controlled. Local IP access (LIPA) and selected IP traffic offload (SIPTO) are considered to be the typical technologies to implement cellular data offloading [7]. They are often discussed in the same context but are different in scope and intent. LIPA is a really part of the femtocell architecture and is designed for use cases such as the residential user who wants to access local services, but doesn’t need to access the wide-area network to do so- for example, to access a film or photo album on the home media computer from a smartphone. SIPTO offloads selected IP traffic to the Internet at home and enterprise environment as well as at macrocellular access networks. The 3GPP working group has recently defined a further study on the standardization on the integration of LTE Direct in mobile networks [8]. They have target use cases to study and identify potential
requirements for an operator network controlled discovery and communications between devices [9] and it comes also along with a feasibility study [10]. In this context, 3GPP technology has the opportunity to become the platform of choice to enable proximity-based discovery and communication between devices, and promote a new set of advance proximity-based applications such; commercial/social use and networking offloading. The 3GPP is targeting the availability of Device-to-Device communications in LTE Release 12 to enable LTE to become competitive broadband communication technologies for public safety networks [11], focusing in its capacity as a new data-offloading solution. Such solution will evolve mobile networks toward a layered topology in which multiple network layers (Femto-network, D2D-network, Wifi-network...) would coexist. This new interest is motivated by several factors, including the popularity of proximity-based services, driven largely by social networking applications, mainly based on the context and proximity information, will offer at the same time the opportunity for operators to extend their mobile network capacities and to alleviate the traffic in their core networks [12]; possible scenarios will be discussed after. Moreover, D2D communications will be an efficient way to extend network connectivity since it will improve the QoS in poor radio coverage areas. In the next sub-chapter, we introduce D2D communications, the main advantages and possible scenarios with all the references to past studies on the possible mechanisms and architectures.

2.2 Device-to-Device communications

First of all we have to define what a D2D communication is; D2D is a technology that allows devices to communicate directly to each other without the infrastructure of access points or base stations, and the involvement of wireless operators. The term “device” here refers to the users who use cell phones or other devices. Different from Machine-to-Machine, D2D is supposed to be an access or link-layer technology, sharing the same resources with the cellular system. The growing popularity of proximity-based services and applications are driving wireless operators to pursue the D2D function in their networks. However, the traditional D2D technologies are inadequate for those purposes. First, the traditional technologies as WiFi or Bluetooth require manual pairing between two devices, causing inconvenience customer usage experiences. The distance of WiFi-direct is claimed to be 16 meters, which means that dozens of devices within the range may be on the list of available pairing devices. Second, the techniques above mentioned are unable to meet the requirements of some applications due to several technical limitations.
Since most of the traditional D2D technologies work in the crowded 2.4GHz unlicensed band, the interference is uncontrollable. Last but not least, the wireless operators cannot make profits using traditional D2D technologies since they work independently without the involvement of the operators. Unwilling to lose the emerging market that requires the D2D function; the wireless operators are exploring the possibilities of introducing the D2D function in the cellular networks as an underlay to LTE-advanced networks.

### 2.2.1 Benefits, usage cases and business models

From a technical perspective, there are many advantages of using a D2D technology [13]. First, due the proximity of the two connected devices, the connection will enjoy high data rates and low end-to-end delay. Second, it is more resource-efficient for proximate D2D user equipment (UE) to communicate directly with each other than routing through an evolved NodeB (eNB) and the core network, direct communications increase spectral efficiency, due spatial reuse through many D2D links. Third, switching from an infrastructure path to a direct path offloads cellular traffic, alleviating congestion, enhances cellular capacity, and provides a better load balancing, and thus benefitting other non-D2D UE as well. As mentioned before, it allows extending the cell coverage area: Figure 2.1 gives an illustration of the potential benefits of D2D.

![Figure 2.1: D2D benefits](image)

The different usage cases and business models of D2D as an underlay to LTE networks have been deeply discussed in last studies [3]. In the next paragraphs we are going to
provide a summary of this information. They made a differentiation to classify all the usage cases: The first category is referred to as peer to peer case, in which source and destination of the data are D2D devices. The second one category is the relay case, which means that the D2D destination device has to relay the exchanged data to a base station which further forwards the information to the final destination device.

**Peer to Peer usage cases**

**Local Voice server:** The D2D technology could be used to offload the voice traffic when the users share the same geographic location. For example, it will be useful in sports or music events.

**Local Data service:** As it is useful for voice traffic, it also could be useful to provide local data service. Some scenarios of D2D communications are illustrated below.

- **Content Sharing:** Two friends sharing photos or videos, a group of students that need to download the subject content during a conference.

- **Multiplayer Gaming:** If you need to play a sports game with a friend or any multiplayer game.

- **Local Multicasting:** Restaurants advertising the daily menu, shops advertising the sale promotion to the passage nearby.

- **Machine to Machine:** Possibility to interconnect a group of devices, e.g. a smartphone connected to a television to display images or videos.

- **Context-aware application:** It is a driving factor for the D2D technologies and is based on the people’s desire to discover their surroundings and communicate with nearby devices.

**Relay usage cases**

**UE as the Gateway to Sensor Networks:** The D2D technology could be useful to convert mobile-phones in internet gateways for many consumer M2M devices, e.g. sensors in home devices such the oven, central heating, or even the car.
**UE cooperative Relay:** This is the case that concerns us and this thesis, a cooperative relay provides a path to download the data from nearby devices and not from the base station, offloading the cellular network.

**Business Models**

From the business perspective, D2D technology should create new business opportunities, but the challenge for mobile operators, is yet to face the threat of Over-the-Top providers who have burst the mobile market with apps that supply instant messaging, multimedia services like photo sharing and video conferencing and other popular services for free. The following are some possible answers to the ”pay for what” question of the D2D communications:
**Pay for Identity:** This will provide the information to link cellular phone number with WiFi or Bluetooth identity, which facilitates D2D connection setup and provides value-added service. The user could search a D2D nearby device by dialing phone number instead of searching for a name.

**Pay for QoS, security and uninterrupted connections:** Use fully controlled D2D technologies for those services which require high QoS through secure and uninterrupted connections.

**Pay for context information:** Operators have deep contextual information about end users and have an emerging opportunity to leverage context to make it pay off competitively.

**Pay for management:** One possible business model is discussed in [14], where non-cellular devices are included in the operator’s subscriber database and automatically associated with the owner’s cellular devices. Each user will have a profile with all the M2M devices associated, with relevant information, authentication/key management and security.

### 2.2.2 Device-to-Device Architecture

As we told before the aim of the D2D technology is to provide a path to communicate two devices without infrastructure, but for the purpose of cellular data-offloading model the base stations are needed. In this subsection we give an overall service model of the cellular data offloading based on D2D communications [15][16]. The operator controls over normal user communication process which mainly lies in four aspects: access authentication, connection control, resource allocation, and lawful interception of communication information. The last aspect is very difficult to achieve for D2D communications, since information is directly exchanged between users bypassing the operator deployed base stations. According to the level of operator control over D2D communications, two categories of operator controlled D2D technologies can be classified [3]:

**Fully controlled D2D Mode**
The D2D link between two User Equipments (UEs) is an integral part of the cellular network, just like the common cellular downlink or uplink connections. The cellular network has the full control over the D2D connection, including control plane functions, e.g. connection setup, maintenance, data plane functions and resource allocation. The D2D connections, share the cellular licensed band with the normal cellular connections. The network can either dynamically assign resources to each D2D connection in the same
way as a normal cellular connection or semi-statically assign a dedicated resource pool to all D2D connections.

**Loosely controlled D2D Mode**
The operators perform the access authentication for the D2D enabled devices. Apart from this, these D2D devices can set up D2D connections and start D2D communication autonomously with little or no intervening from the operators. To avoid interference to the normal cellular users, the D2D communications can make use of either the unlicensed band with WiFi or Bluetooth for data transmission or a dedicated carrier on the licensed band.

In both cases following D2D specific functionalities are supported.

- **Proximity measurement**: The UE can send/receive a proximity measurement signal from the eNB via LTE-Uu interface.

- **D2D channel state measurement**: The UE can send/receive a D2D channel state measurement signal from the eNB via LTE-Uu interface.

- **D2D data transmission**: Data transmission between devices is performed using D2D physical shared channels.

As shown in Figure 2.3, UE1 and UE2 are engaged in a cellular data communication that is being routed over the eNB and the core network infrastructure. The gateway is able to detect potential D2D traffic since it actually processes the IP headers of the data packets. After detecting D2D candidates (UE1 and UE2), the gateway informs the candidate UEs that they have to perform a discovery process. The discovery is made by one party transmitting a known reference signal sequence. At that time, the serving eNB coordinates that the two peer UEs meet in space, time and frequency with the reference signal. As the UE1 and UE2 come into proximity of each other, each detects the peer in its proximity. Then the cellular data session between UE1 and UE2 is switched to a D2D communication path. Later, at some point, the D2D data session is switched back to the cellular path when the D2D communication path is no longer feasible. In the proposed data-offloading model [17], the user will not perceive the switching of user traffic sessions between the cellular communication and D2D communication paths.
2.2.3 Device-to-Device communication procedure

The procedure of a D2D communication can be divided into three phases, the device discovery, where the devices have to detect the presence of other devices in the neighbourhood, the link setup phase, where the devices establish links between them, and finally the data communication, where the data is transmitted.

Device Discovery
Existing work can be classified into centralized and distributed approaches [3, 13]. As shown in Figure 2.3, the discovery could be done either directly between D2D devices, or using the base station.

Centralized approach: The base station or a certain entity in the cellular network detects that it may be better for two communicating UEs to set up a D2D connection. This entity, then informs the eNodeB to request measurements from the UE to check if the D2D communication offers higher throughput. If so, the eNodeB decides that the two UE can communicate in D2D mode [16]. The control of the discovery phase at the core network ensures the delivery of a better QoS and a more reliable service, thanks to

![Figure 2.3: Data offloading model](image-url)
authentication and privacy mechanism offered natively by the operator. However, this approach could be less scalable than the direct approach as it could have performance and overload impact on the core network with the additional D2D traffic, also, assuming synchronization a priori before device discovery may be questionable in the out-of-coverage scenario.

**Distributed approach:** The UE broadcasts identity [18] periodically so that other UEs may be aware of its existence and decides whether it shall start a D2D communication with it. This approach does not need the involvement of the base station [19]. However, such a method may have an impact on the UE complexity as it needs to support power management mechanism to avoid battery consumption issues causes by an “always-on” discovery.

**Link Setup**

When the discovery phase is done, D2D peers establish a communication link for data exchange. Basically, in the case of a D2D communication, a dedicated D2D bearer with specific radio resources allocated by the eNB is setup. According to the current 3GPP standard on ProSe [9, 10], direct D2D communications are allowed only when devices are out of coverage and only for Public safety services and bearer mechanism are not yet defined for these types of communications. In [17], a D2D dedicated bearer mechanism is proposed for a D2D offloading service. Generally, existing D2D studies, lack of detailed mechanisms for D2D bearer establishment for D2D-based services other than public safety and offloading.

2.2.4 Related Work

In this sub-chapter, we are going to analyse and summarize some previous works that have been done in data-offloading field that has a relevant importance in this thesis. The first proposed peer-to-peer mechanisms were swarming protocols such as BitTorrent [20], Slurpie [21] or Dandelion [22]. Swarming is a peer-peer content delivery mechanism that utilizes parallel download from the WiFi AP to divide the full content among a mesh of cooperating peers, so that all of the users share the content pieces to end up having the full content. In [23] it is investigated how D2D communication can be integrated with cellular networks and the applications it can support. The problem of resource allocation is also studied in [24, 25]. Also, there are papers referring to the power control
in D2D communications [3]. The interference coordination with the cellular network is studied in [16]. The article [17] investigates the concept of the D2D discovery and the D2D pair and identifies the protocol architecture to adapt these concepts. In the article [26] it has addressed some topics including interference management, multi-hop D2D communications with network coding; mechanism such as mode selection, resource allocation, D2D communications with multi-antenna transmission techniques and power control are illustrated in detail. The D2D communications in heterogeneous networks are discussed in [27] where is explained a solution to the problem of selecting which endpoint should serve a user, and the radio resources to allocate for such communication, the results of those investigations conclude that D2D mode can be a valid, low-cost alternative to microcells in supporting traffic with little energy consumption. For the purpose of this thesis, where we are characterizing the downloaded content based in real HTTP traces from the Politecnico di Torino cellular network, some theoretical studies have been done [28] that prove through some paradigms and models that, even in such a highly dynamic setting, a relatively low user density is enough to guarantee content persistence over time. Once is declared that D2D communication is a possible mechanism, and there is content persistence over time, an analysis of the content is needed to see if it is possible to use this floating content in Device-to-Device for cellular data-offloading. The cacheability of the content is studied in [29] where the potential of forward caching in 3G cellular networks is demonstrated. Then in [30] they found that content prestaging, by proactively and periodically broadcasting “bundles” of popular object to devices, allow both greatly, improving users’ performance, reducing up to 20% (40%) the downloaded volume in optimistic scenarios with a bundle of 100MB. And finally in [31] is studied a comparative of Cellular Multicast and Device-to-Device communications concluding that when content dissemination tolerates some delay, providing device-to-device communications over an orthogonal channel increases the efficiency of multicast.
3 - Traffic analysis

3.1 Data Collection

The traffic traces that are used in this thesis have been collected by a free open source tool called Tstat (TCP Statistic and Analysis Tool). Tstat is a passive real time monitoring tool that can extract much useful information from the observation of the packets. The Tstat is placed right after the Edge router that carries all the packets coming and going from/to the users of the Polito network, with this setup is possible to extract the desired information. For example, using the two couples Client/Server IP addresses and TCP ports is possible to determine a flow connection. Then, it can classify all these information obtaining statistic about incoming and outgoing traffic. Tstat generates different types of measurement collections. For our study we are going to focus just in the Log files, storing flow-level measurements.

Figure 3.1: Tstat setup
3.2 HTTP logs

The Hypertext Transfer Protocol (HTTP) [32] is designed to enable communications between clients and servers; it works as a request-response protocol. For example, a web browser may be the client, and an application on a computer that hosts a web site may be the server.

![HTTP protocol flow](image)

For the purpose of this thesis, we collected HTTP requests and responses headers at the interface in Figure 3.1 in the wireless network in Politecnico di Torino over different periods. During each one of this period, it can be seen thousands of UEs including laptops, smartphones and tablets making thousands of HTTP requests. As our traces come from the Polito WiFi network, the majority of our traffic is coming from computers, this could be a problem since we want to study the D2D offloading infrastructure, but since smartphones and tablets are becoming more popular and people use them to download the same content as they did with computers, we can still take relevant information from these traces. The HTTP logs are stored as TXT files, where each row corresponds to a different flow and each column is associated with each information measure. The first two rows provide a description of each field of each column.

![HTTP header fields](image)

As we can see from the Figure 3.3 the first row refers to the client-to-server flows while the second one to the server-to-client counterpart. Both types of flow share the same first five parameter fields:

1. Client IP address
2. Client TCP port
3. Server IP address
4. Server TCP port
5. Instant time, as Unix time
The Client IP addresses present in the trace are anonymised, but it is still possible to associate to the same user, even if it is anonymous, those requests coming form the same client; this is very useful to characterize the data, as it will be explained in the chapters below.

From here, the client-to-server flow has five more fields:

6. Method  
7. Hostname  
8. fqdn  
9. Path  
10. Referer  
11. User Agent

The two common methods for a request-response between a client and server are GET (requests data from a specified resource) and POST (submits data to be processed to a specified resource). With the hostname + path it is possible to identify all the different objects that are present in each trace. The fully qualified domain name (FQDN) is the original server hostname the client resolved. The referer is the address of the previous web page from which a link to the currently requested page was followed. And finally from the User Agent it is possible to identify which type of device has been used.

Considering the server-to-client flow has seven more fields:

6. Method: is fixed as HTML  
7. Response  
8. Content-len  
9. Content-type  
10. Server  
11. Range  
12. Location

The response field is important to identify if the object been requested has been actually served by the server. To proceed with the analysis of the traces I just have taken in count those whose response was equal to 200, implying that the request has succeeded. The next field is content-len which provides the actual size in bytes of the object; this field is not always filled by the server so as it will be explained in the chapters below, we have to be careful about this field. It happens the same with the content-type field, the internet media type of this content. As it is filled by the server, sometimes the content-type do
not match with the correct type of content. The server field identifies the type of server that has response the request, many times is not filled. The range point up where in a full body message this partial message belongs and location field to ask a web browser to load a different web page.

### 3.3 APs logs

To perform the Spatial Correlation analysis we needed information about the users’ spatial distribution. We got this information from the APs traces; we could extract how many users were connected to one single AP at a certain time, making possible to perform a Time Correlation analysis for each AP instead for the whole net. The aim of this part is to find if users can download the content from neighbouring users under the same or nearby APs. The APs logs are stored as CSV files, where each row corresponds to a different connection and each column is associated to each information measure.

<table>
<thead>
<tr>
<th>Client IP Address</th>
<th>Client MAC Address</th>
<th>Association Time</th>
<th>AP Name Map Location</th>
<th>SSID</th>
<th>Protocol</th>
<th>Avg. Session Throughput (Kbps)</th>
<th>RSSI</th>
<th>Session Duration</th>
<th>AP Radio</th>
<th>Session ID</th>
<th>Association ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Client IP address 5. Map Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Client MAC address 6. Protocol</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Association Time 7. Disassociation Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. AP Name 8. Session Duration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As it can be seen from Figure 3.4 there is much information that can be extracted from the Trace. For the purpose of our thesis, we are going to focus in the next fields. With the Client IP and the Client MAC address we can differentiate each user connected. The Association Time, Dissassociation Time and Session Duration provide timely information to know the exact time the user was connected to the AP. The Map Location and the AP Name give the exactly AP and the exact location to perform the analysis, and to compare the nearby APs.
4 - Data Characterization

In this chapter it is shown the meaningful information that can be collected from the traces and the parameters that can be obtained. First of all, we are going to define some concepts that are needed to understand the analysis. For the following definitions and explanations we have used traces corresponding to the day 30/11/2014 at 20:45 GMT.

4.1 Object Definition

For the purpose of this thesis, we have to define what an “object” is. We consider that an object is any single item that can be downloaded from the internet: images, videos, html pages, text, Java scripts, executable files, etc. In order to statistically analyse these objects is necessary to find a form to be able to distinguish each object from the others. As it is possible to extract the hostname and the path of each downloaded item from the trace, we will concatenate those strings to obtain the unequivocal reference of each object. Copying the whole concatenated string into any address bar of an internet browser will lead to the content. For example, the LinkedIn logo located on the Polito main website has an address which is split and stored in the log files as:

3. Hostname: www.polito.it

4. Path: /images/linkedin.png

Figure 4.1: LinkedIn Logo
In the Figure 4.2 is shown the twenty-five more requested objects in the trace. It can be seen that all of them share the same hostname: the www.polito.it. That is because the trace was from a Sunday, when the university was closed, so all the requests are from outside users downloading content from the Polito net. In a working day, we will see much more diversity in the content. Even we are analysing traces in the Polito; the most popular objects are from well-known sites as Facebook, Google and related to specific operating system updates.

![Figure 4.2: Twenty-five more requested objects](image)

### 4.2 Request Definition

Once we are able to distinguish each object, it is basic to characterize the information that we are going to analyse in this thesis. To determine the possibilities of the D2D offloading it is necessary to find out if there is a correlation between the content that different users download through their mobile devices. We will analyse some concepts; the popularity of the objects, as the number of users that request each object and the number of requests for each object; the size of each object and the type of content. This being said, it is essential to define what a request is; a single object gets a request when it is asked for by a user, meaning that an HTTP request and an HTTP response are generated. Those requests are collected in the log files. Afterwards, those two lines are
unequivocally coupled together by a script that compares client and servers’ IP addresses and TCP ports in order to obtain the statistical data. With the IP address we are able to distinguish each different user present in the trace.

### 4.2.1 Popularity

There are two approaches to the popularity concept. An object could be popular because the number of requests from many different users 4.3(a), or because the number of requests including more than one request for each user 4.3(b). There is a need for two definitions because when we were analysing the data taking into account the second approach, we noticed that some contents a priori not so popular, were placed in the top 100 requested objects. So, by filtering all the IP addresses that asked for them, it turned out that those contents were requested by only one user, hundreds of times. To remove this inflation, we decide to take the first approach as the reference because in a theoretic scenario a user will request each object once; in the second definition we will count biased requests due to the possibly wrong functioning of some users’ devices or due to malware. This also helped to reduce another inflation, generated by the auto-refreshing items on the websites, like the ads that appear in many popular sites. This policy may not consider if a client actually asks for a specific item more than one time, but the frequency at which this case happens is so low that it makes the inaccuracy negligible.

The script parses the log file creating a table of all the different objects from the trace to keep track of the total amount of users that has requested it. As it can be in Figure 4.3, the one-hundred most requested objects are ordered by the number of users’ requests they have, meaning that the most requested will be in the first position of the abscissa axis, this order will be followed in all the plots, because it is interesting to keep tracking each object along the study, to be able to get some correlation along the plots; so the object #1, will be the same in all the different plots.

In the Figure 4.3(b), it is shown the second approach to the popularity; here it is shown the number of requests of each object independent from the number of requesters, there is no distinction between request coming from different users or from only one of them. As it can been seen, it has the same behaviour that Figure 4.3(a) but there are some peaks due to more than one request by each user; for example, object #58 has been requested by one user many times. From both plots it seems that the popularity decreases quite fast, but there are only 100 objects present while there are, in general, two to four millions of different objects in a single hour log. To get a more clear view of this behaviour, we plot
the Number of Request but for a bigger set of users. In the Figure 4.4 is shown the hits of 10,000 objects. With this small fraction is sufficient to show how fast the popularity decreases, the number of hits is heavily unbalanced between the first objects and the rest of them.

This graph is following a behaviour described by the so called Heavy-tailed distributions. In probability theory, these distributions are probability distributions whose tails are not exponentially bounded [33]: that is, they have heavier tails than the exponential distribution. There are three important subclasses of heavy-tailed distributions: the fat-tailed distributions, the long-tailed distributions and the subexponential distributions. In our
study, the curves follow the Pareto distributions, which belongs to the subexponential category. Its probability density function (PDF) is:

\[
 f_x = \begin{cases} 
    \frac{\alpha x^\alpha}{x_m^\alpha + 1} & x \geq x_m \\
    0 & x < x_m 
\end{cases}
\]  

(4.1)

Here, the \( x_m \) is called scale parameter, while the \( \alpha \) coefficient is the shape parameter, also known as tail index. Fixing the scale parameter equal to 1, the contribution of the \( \alpha \) can be seen in Figure 4.5. As \( \alpha \) increases, the distribution approaches \( \delta(x - x_m) \), where \( \delta \) is the Dirac delta function.

Figure 4.5: Pareto probability density function for various \( \alpha \) with \( x_m = 1 \)

Finally, we will show the popularity distribution of the objects, in percentage. Normalizing the hits count, dividing it by the total amount of hits present in the log file leads to a percentage value which represents the probability of an object to be hit each time a request is made. As shown in Figure 4.6 adopting the logarithmic scale, even if the likelihood of the first few objects is much higher than the others, it is still a very low number, due to the massive quantity of different contents. And taking into account that we are analysing a trace from a Sunday, if we do the same analyse for a working day, i.e. a Wednesday, we can see this numbers are far lower. This means that our most requested items, those that we called the popular ones, are not so popular indeed.
4.2.2 Size

In this subsection we are going to analyse the size of the objects studied. It is very important to characterize the objects in terms of size. The D2D offloading technique would require that every device allocate part of their memory to cache, this means that size could be a determinant factor for the reliability of our system. As far as we know, as it can be seen at Figure 4.7(b), the majority of objects present in the trace are in the range from 0B to 100B, and the rest is more or less uniformly distributed from 100B to 2.49GB. But this statement does not imply that the most popular objects are small. As it can be seen from Figure 4.7(a), where is shown the size of the one-hundred most requested objects, we cannot extract any conclusion about the size of the most popular objects. This is not a surprise since in the one-hundred most requested objects we can find images in the order of KB and html pages that are in the order of Bytes. But it is true, that as we will explain in the next sub-section, we are just interested in those content that are strictly static, discarding those that are mostly bigger, such as videos and applications.

To check the size of the object it has been used a script that parses each object with his content-length field. We had to be very careful with this analysis because this field is filled by the server, and many times is not able to find out and it fills the value with a ‘-‘. To build the size plots for the analysis of the one-hundred most requested objects we had to download all the none-size objects to check the real value and replace it. For all
the results extracted from analysing more than one-hundred objects, we used a script to discard all those objects that were 0 bytes, or unknown. When we were analysing the data, we also found that the same object often appears with a different size value for each request. Due to that, three different scenarios can be extracted, Best-Case where all the different size values have been merged in the higher value. The Average-Case where the selected value is the average of all the different size values present in the trace and finally the Worst-Case, where for each different size value a new object has been defined.

**Figure 4.7: Size analysis**

(A) Size of the 100 most requested objects

(b) PDF of the size of the 100 most requested objects
4.2.3 Type of content

The type of content of each object is one of the most important characteristics to look at it, because even if an object is very popular, to make D2D offloading technic useful, the user has to be able to download and store it to share it with other users. Some concepts have to be presented in this section. An object is cacheable if it is possible to download and store it, the cache hit ratio for the overall population of UEs is 33% [29]. For the purpose of this thesis, we are going to classify objects in two classes: static and dynamic objects. We are going to assume that static objects are cacheable, in spite of the fact that some content are forbidden to be cached by their providers. A static object is a website where the content is always the same; it is served up to any visitor the same. On the other hand, in a dynamic website the content on the server is the same, but instead of just being HTML, it also contains dynamic content, which may display different data depending on information such as the time of day, the user who is logged in, the date, etc. To analyse the traces we used a script that parses each line of the log file and checks in the content-type field, but it happens the same that in the content-len field, as it is filled by the server it is usual to find errors in this field, so we would have to manually check it. For all the results extracted from analysing more than one-hundred objects, we used a filter to discard all those objects that are clearly dynamic, those that contain the words “.php” “.shtml” “.asp” “/api/”. As it can be seen at Figure 4.8 we could extract content type information from the traces:

![Type of content](image)

(A) Type of content

![Static type of content](image)

(B) Static type of content

**Figure 4.8:** Type of content of the 100 most requested objects.
5 - Time correlation

D2D offloading technique is a cooperative mechanism where users take advantage of the downloaded content by their neighbours to avoid connecting through the router. Trying to extract some time correlations among different users through consecutive times is primordial to see the reliability of the mechanism. The traces used in this analysis were collected the first week of February 2015. We started by taking three different days; Monday, Tuesday and Wednesday. Analysing them separately could give information about the behaviour of the users among different days. For the purpose of this analysis the time interval is 11:42-12:42, because it is a period of the day where many students are moving around the Politecnico di Torino. This analysis considers the whole Polito as one single net, not taking into account that two users should be under close APs to share content.

5.1 Consecutive days analysis

In this first sub-section of the Time Correlation analysis we are going to analyse if the characteristics of the content are stable during the days, i.e. the most requested object in day 2 and day 3 have been requested by the same similar number of users. We started by looking the Number of Users that has requested each object in the three different days, Figure 5.1-5.2-5.3(a), as we told in the chapter 4, this is the first definition of Popularity, and the reference one. From the Number of Users plot of each day, it is visible that in the three of them the most requested object is requested by around 2,000 different users. And the three curves are following more or less the same behaviour. That means that the one hundred most requested objects are requested every day by the same similar number of users. It is also visible that every day, just 6 objects are requested by more than one-thousand different users. As there are seven-thousand users present in the trace, is sufficient to show that the most requested objects are not so popular indeed. Then it is
also visible the decrease between the first most requested object and the one-hundredth. The one-hundredth has been requested just 203 times, as all the different objects present are 1,568,364 it is easy to see how fast the popularity decreases; as it was explained before, the number of hits is heavily unbalanced between the one hundred first objects and the rest of them. From the Number of Users plot, Figure 5.1-5.2-5.3(b), we can conclude that the one hundred most requested objects have similar characteristics in three consecutive days, this is probably because in consecutive days, most of the objects are repeated, this will be analysed in the next sub-section. If we look at the Number of Request plots, not much information can be extracted apart from the previous explained, as expected it follows the same behaviour as the Number of Users plot, the differences are due the inflations explained in the previous chapter: malware, auto-refreshing items etc. In day 2 there is a clear example that this second definition leads to incorrect assumptions, we can see that object #99 is the second most requested object, so if we look at the IP that has requested this object we find that two users have requested this object hundreds of times. For example, if we take a look to the object #40, that is the www.msn.com main page, we find that this object had been requested by 412 different users, but if we take a look the number of requests that had received we find that it had been requested 2,342 times. This difference could lead the analysis to wrong conclusions. The Size plot is not giving much information, Figure 5.1-5.2-5.3(c), doing a quick view we cannot conclude that the most requested are smaller or bigger than the less requested. What is visible is that even we are taking into account dynamic content, which is usually bigger than static content, in the one-hundred more requested there are not objects above 100KB, which is important in terms of memory allocation. We also plot the Traffic Volume, Figure 5.1-5.2-5.3(d), build multiplying the number of request by the size of each object, to show that not the most requested object are the responsible for the highest amount of traffic volume. Indeed, this plot reflex the importance in taking into account the size of the objects into the study, it could be possible that for a better performance to offload the net is more important to store a low requested object but bigger in terms of size, because will lead to bigger savings in offloading the cellular net. Finally, we can compare the Type of Content present in each day, Figure 5.1-5.2-5.3(e)(f). It is noticeable that the types of objects present in three consecutive days are quite stable, we found that a bit less than 70% of the one hundred most requested objects are static, and those are divided in 80% (text) and 20% (images). This is a good new, since D2D offloading technique is just interested in Static Content, which can be shared between users independently of who have requested it. Now that we have a view of the three days independently, we can compare them. In the next sub-section will be shown that these similarities are due
to that most of the one-hundred most requested objects are repeated during consecutive days.

**DAY 02-FEB-2015**

**Figure 5.1:** Day 02-Feb-2015
Figure 5.2: Day 03-Feb-2015
Time correlation

**DAY 04-FEB-2015**

![Graph](image1)
(A) Number of Users

![Graph](image2)
(b) Number of Requests

![Graph](image3)
(c) Size

![Graph](image4)
(d) Traffic Volume

![Graph](image5)
(e) Type of Content

![Graph](image6)
(f) Static Type of Content

**Figure 5.3:** Day 04-Feb-2015
5.1.1 Comparative of day 2 and day 3

In this sub-section we are going to compare two consecutive days. The aim of this analysis is to extract the savings that would achieve the D2D offloading mechanism if users from day 2 could share their downloaded objects with the users connected the following day, at the same time. So, for example, if a user in day 2 had downloaded the Politecnico di Torino logo from the www.polito.it website, and a user in day 3 asks for the same object, instead of downloading from the network, would download it directly from the user of day 2. In the Figure 5.4 is shown the comparison between day 2 and day 3 at 11:42 to 12:42 taking as reference day 2, which means that objects requested in day 3 not present in day 2 are not shown. In this plot we can see objects present in day 2 in blue, and the objects present in day 3 in red, each one with his value in terms of Number of Users and his corresponding Size in grey. It is visible the level of repetition between two consecutive days, and the slight differences in terms of requests. It is also observable that some day 3’s points are located on the x-axis; which means that those objects present on day 2 are not present in day 3. Looking into those objects, we found that the three first ones were certificate revocation lists (crl), so probably those lists are periodically published on Monday. The rest of the objects placed on the x-axis probably don’t appear on day 3 because we are just comparing the one-hundred most requested objects, if we take into account a larger set of objects, we will see that those objects probably were also requested in day 3. So as it is explained before, the most popular objects are not so popular indeed, but they are constantly popular through consecutive days.

![Figure 5.4: Comparison day 2 and day 3](image-url)
In the next Figure 5.5 is the same comparative but taking day 3 as reference. As it is visible, the behaviour is stable, the crl lists has disappeared and the not-repeated objects are located at the end of the plot, which means that the differences increase with the increase of the objects analysed. As the most popular objects are well-known contents coming from Facebook, Google, Twitter, and related to specific operating system updates and certifications it does not comes as a surprise the repetition of those objects in consecutive days.

![Figure 5.5: Comparison day 3 and day 2](image)

A further step in the analysis is to see the behaviour and the savings when considering a larger set of objects. We build the tables and graphs when considering 500, 1000 and 2000 objects, taking into account the three scenarios based on the different size considerations; The Best-Case where all the different size has been merged in the higher value, the Average-Case where the selected value is the average of all the different sizes, and the Worst-Case where for each different size a new object has been defined. The objects repeated are those that were requested in both days, so if the D2D technology were implemented, all the requests will be saved except the first one, and the traffic is computed multiplying the number of hits by the size of each object.

As it can be seen in Figure 5.6 the number in percentage of objects repeated through days decrease with the increase of objects analysed. We can see that the objects repeated and the requests saved do not change between the Best-Case and the Average-Case scenario, because as we explained before in these two different scenarios we are changing the way to define the size for each object, but we are not defining new objects depending on the
size. In the Worst-Case where, to count an object as “saved”, it must be in both days, with the exactly same size and path. The differences between the three scenarios give us an idea of how many objects have the same name but they have a different size value. When we are considering the Worst-Case, which is the more realistic one, we are saving half of the traffic saved in the other two scenarios. From Figure 5.7 is visible that while the Best-Case and Average-Case follows the same behaviour, the Worst-Case is far lower.

Besides those differences, the content downloaded has a great level of “repeativeness”. Even in the Worst-Case with 2,000 objects analysed we are saving the 55% of the objects and the 40.87% of the Traffic.

<table>
<thead>
<tr>
<th>100 objects considered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario</strong></td>
</tr>
<tr>
<td>Best-Case</td>
</tr>
<tr>
<td>Average-Case</td>
</tr>
<tr>
<td>Worst-Case</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>500 objects considered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario</strong></td>
</tr>
<tr>
<td>Best-Case</td>
</tr>
<tr>
<td>Average-Case</td>
</tr>
<tr>
<td>Worst-Case</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1000 objects considered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario</strong></td>
</tr>
<tr>
<td>Best-Case</td>
</tr>
<tr>
<td>Average-Case</td>
</tr>
<tr>
<td>Worst-Case</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2000 objects considered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario</strong></td>
</tr>
<tr>
<td>Best-Case</td>
</tr>
<tr>
<td>Average-Case</td>
</tr>
<tr>
<td>Worst-Case</td>
</tr>
</tbody>
</table>

**Figure 5.6:** Comparison table day 2 and day 3
Figure 5.7: Comparison graphs day 2 and day 3
5.2 Consecutive intervals analysis

D2D data-offloading technic is a real-time mechanism, which means that users take advantage of the content downloaded by neighbouring devices. Moreover, provider and requester have to be connected at same time in order to be able to share the content. This need to coexist in time and space force us to perform a more accurate observation in time. In this sub-section of the Time Correlation analysis, we are going to study different ten minute consecutive intervals, to extract the savings that would achieve this mechanism if users from one interval could share their content with the users of the following interval. In order to perform this analysis, we took the log file corresponding to 02-Feb-2015 from 11:42 to 12:42 and we split the hour in ten minute intervals as it is shown in Figure 5.8.

<table>
<thead>
<tr>
<th>Interval</th>
<th>Initiate</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval 1</td>
<td>11:42</td>
<td>11:52</td>
</tr>
<tr>
<td>Interval 2</td>
<td>11:52</td>
<td>12:02</td>
</tr>
<tr>
<td>Interval 3</td>
<td>12:02</td>
<td>12:12</td>
</tr>
<tr>
<td>Interval 4</td>
<td>12:12</td>
<td>12:22</td>
</tr>
<tr>
<td>Interval 5</td>
<td>12:22</td>
<td>12:32</td>
</tr>
<tr>
<td>Interval 6</td>
<td>12:32</td>
<td>12:42</td>
</tr>
</tbody>
</table>

**Figure 5.8:** Time intervals analysed

5.2.1 Comparative Interval 1 and Interval 2

This first plot, Figure 5.9, shows the comparison between Interval 1 and Interval 2; taking as reference the Interval 1. As expected, as we are considering a shorter time interval and consequently a less number of users, it is visible that the one hundred most requested objects received less hits than in the previous analysis. From this plot it can be seen that the most requested objects are repeated trough the consecutive intervals, but is noticeable the dispersion in popularity between them. Also, we can see that objects are in the range of 10B to 100KB. Next step in the analysis is considering a larger set of objects. If we look at Figure 5.11 it is remarkable how the dispersion increases as long we increase the number of objects studied, is visible that the one hundred most requested object share more or less the same amount of hits in the two intervals, but as we consider more objects, the differences increase. From the plot it is also visible than when we are considering more than 500 objects, the number of objects repeated decrease. However, even we
consider more objects in our study; we got a high value in terms of “repeativeness”. As we are comparing consecutive time intervals, this level of “repeativeness” does not come as a surprise, but we can conclude that the number of objects repeated will decrease for a larger set of objects analysed as will for a larger time intervals.

Figure 5.9: Comparison Interval 1 and Interval 2 of 100 objects

Figure 5.10: Comparison Interval 1 and Interval 2 of 500 objects
Figure 5.11: Comparison Interval 1 and Interval 2 of 1000 objects

Figure 5.12: Comparison Interval 1 and Interval 2 of 2000 objects
The next step is to see the savings when considering a larger set of objects but taking into account the three scenarios based on the different size considerations. It is visible that the level of “repeativeness” is higher than when we analysed Consecutive Days. Even when we are considering 2,000 objects in the worst case we have the 54% of the objects repeated, which means that D2D data-offloading mechanism would save the 44% of the traffic. From Figure 5.15, we can see the differences between the three scenarios, and how, as we take into account more objects in the analysis the increment in Traffic Saved and Request Saved is becoming lower.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Objects Repeated</th>
<th>Objects Repeated (%)</th>
<th>Requests Saved</th>
<th>Traffic Saved (Bytes)</th>
<th>Traffic Saved (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best-Case</td>
<td>89</td>
<td>89%</td>
<td>9813</td>
<td>1.43970E+08</td>
<td>97.08%</td>
</tr>
<tr>
<td>Average-Case</td>
<td>89</td>
<td>89%</td>
<td>9813</td>
<td>1.10487E+08</td>
<td>95.87%</td>
</tr>
<tr>
<td>Worst-Case</td>
<td>83</td>
<td>88%</td>
<td>6014</td>
<td>6.16466E+07</td>
<td>95.24%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Objects Repeated</th>
<th>Objects Repeated (%)</th>
<th>Requests Saved</th>
<th>Traffic Saved (Bytes)</th>
<th>Traffic Saved (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best-Case</td>
<td>333</td>
<td>77%</td>
<td>16441</td>
<td>2.40897E+08</td>
<td>92.88%</td>
</tr>
<tr>
<td>Average-Case</td>
<td>333</td>
<td>77%</td>
<td>16441</td>
<td>1.71061E+08</td>
<td>86.25%</td>
</tr>
<tr>
<td>Worst-Case</td>
<td>333</td>
<td>69%</td>
<td>10233</td>
<td>1.05138E+08</td>
<td>86.41%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Objects Repeated</th>
<th>Objects Repeated (%)</th>
<th>Requests Saved</th>
<th>Traffic Saved (Bytes)</th>
<th>Traffic Saved (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best-Case</td>
<td>712</td>
<td>71%</td>
<td>20098</td>
<td>2.85462E+08</td>
<td>86.09%</td>
</tr>
<tr>
<td>Average-Case</td>
<td>712</td>
<td>71%</td>
<td>20098</td>
<td>2.24389E+08</td>
<td>83.04%</td>
</tr>
<tr>
<td>Worst-Case</td>
<td>606</td>
<td>61%</td>
<td>12552</td>
<td>1.27124E+08</td>
<td>60.53%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Objects Repeated</th>
<th>Objects Repeated (%)</th>
<th>Requests Saved</th>
<th>Traffic Saved (Bytes)</th>
<th>Traffic Saved (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best-Case</td>
<td>1346</td>
<td>67%</td>
<td>25222</td>
<td>3.86868E+08</td>
<td>71.30%</td>
</tr>
<tr>
<td>Average-Case</td>
<td>1346</td>
<td>67%</td>
<td>25222</td>
<td>2.88431E+08</td>
<td>64.71%</td>
</tr>
<tr>
<td>Worst-Case</td>
<td>1089</td>
<td>54%</td>
<td>15538</td>
<td>1.57544E+08</td>
<td>44.04%</td>
</tr>
</tbody>
</table>

**Figure 5.13:** Time intervals table
Figure 5.14: Comparison graphs Interval 1 and Interval 2

<table>
<thead>
<tr>
<th></th>
<th>BEST-CASE</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Interval</td>
<td>Objects Repeated</td>
<td>Request Saved</td>
<td>Traffic Saved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-&gt;2</td>
<td>89</td>
<td>9929</td>
<td>1,435,494E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2-&gt;3</td>
<td>89</td>
<td>9812</td>
<td>1,439,705E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3-&gt;4</td>
<td>91</td>
<td>9840</td>
<td>1,435,598E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4-&gt;5</td>
<td>89</td>
<td>9270</td>
<td>1,378,964E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-&gt;6</td>
<td>90</td>
<td>9624</td>
<td>1,341,951E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>89</td>
<td>9812</td>
<td>1,439,705E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>89%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>AVERAGE-CASE</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Interval</td>
<td>Objects Repeated</td>
<td>Request Saved</td>
<td>Traffic Saved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-&gt;2</td>
<td>89</td>
<td>9929</td>
<td>1,070,585E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2-&gt;3</td>
<td>89</td>
<td>9813</td>
<td>1,104,874E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3-&gt;4</td>
<td>91</td>
<td>9846</td>
<td>1,133,005E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4-&gt;5</td>
<td>89</td>
<td>9270</td>
<td>1,120,386E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-&gt;6</td>
<td>90</td>
<td>9624</td>
<td>1,072,315E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>89</td>
<td>9813</td>
<td>1,104,874E+08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>89%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(A) Objects Repeated

(B) Objects Repeated (%)

(C) Traffic Saved
Figure 5.15: Comparison graphs Interval 1 and Interval 2
6 - Spatial Correlation

As it has been explained before, the D2D offloading technique is a cooperative mechanism where users take advantage of the downloaded content by their neighbours to avoid connecting through the base station. In this chapter, we are going to take into account the spatial distribution of the users in the Politecnico. In the previous chapters we assumed that the whole Politecnico was a single net, where all the users could share content between them. This assumption is far from being true, in reality, only users who share the same AP or nearby AP could share the content. Therefore, we are going to analyse the behaviour of users in each AP, extracting how many requests will be saved if the D2D offloading technique was implemented in different scenarios: one single AP, taking in count one AP and his neighbours etc.

First of all, we need to explain some constraints we encountered to perform the analysis. On March 2015 the supervisor of the Polito WiFi network decided to replace the routers for a new model. When some of the new routers were installed, they started capturing just traces from the new network. This led us to only be able to extract information from the new routers that were placed in the Atrium of the Politecnico, Figure 6.1. This means that we were not able to get the AP traces corresponding to the HTTP trace studied previously, forcing us to make some assumptions to proceed with the analysis. For each different AP, we extracted the different MAC addresses connected to it, then we extracted randomly the same number of IP addresses from the HTTP trace and we matched them randomly. With this procedure we were able to distribute the content information among the different users connected. Furthermore, to decide the neighbouring AP we needed to perform a field test to get the exact coverage area and the corresponding data rate for each position in the Atrium, we didn’t have time to do it, so we assume that only the one-hop far away APs were neighbours.
6.1 Users under the same AP

The aim of this part of the work is to understand the performance of the D2D technique in one single AP. We started the analysis by choosing between the different APs located in the Atrium, we decided to begin with the AP-DITAG01 because it was the one with a higher number of users connected to it. We stored the MAC addresses and the corresponding Association/Disassociation Time for each connected device. Then we extracted randomly the same number of IP addresses from the HTTP traces and we matched them randomly. In addition, we plotted the connection time of each device to get a clear view of the time correlation between devices, Figure 6.2. It is visible that the moment with more users connected was from 12:45 to 13:00, when the lessons finish at lunchtime. We will expect a better performance in that interval due the higher number of users connected.

To proceed with the analysis, we used a script that stores each IP address and his corresponding connection time and parses the HTTP log to find if their requested objects were previously downloaded by someone connected at the same time to the AP; if there is, we count this object as “Cached”, if not, this object would be downloaded from the network. It is important to highlight that is indispensable that both users must be connected at the same time and the provider must have downloaded the content before the requester asks for it. The results for the whole hour are shown in the table Figure 6.3.
The number of object cached is remarkable since the D2D technique has saved the 43% of the traffic in a single AP, remember that we are considering a best-case scenario because we are not discarding those object requests made by the same user. To get a more clear view of the performance of the D2D technique we divided the one-hour interval in 15 minute intervals, Figure 6.4, this will show the differences between the peak time and the off peak time, and the average of objects cached during an hour. The results are shown in the table Figure 6.5.

<table>
<thead>
<tr>
<th>Interval</th>
<th>Initiate</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval 1</td>
<td>12:03</td>
<td>12:18</td>
</tr>
<tr>
<td>Interval 2</td>
<td>12:18</td>
<td>12:33</td>
</tr>
<tr>
<td>Interval 3</td>
<td>12:33</td>
<td>12:48</td>
</tr>
<tr>
<td>Interval 4</td>
<td>12:48</td>
<td>13:03</td>
</tr>
</tbody>
</table>

As expected, we can conclude that the number of requests saved increases as the number of users connected. The interval 3, which start at 12:33 and finish at 12:48, is the peak time, with more users connected to the AP and thereby the moment when the D2D technique presents the best performance.
6.2 Users under neighboring APs

In this subsection we are going to analyse the performance of the D2D technic taking into account the neighbouring APs. For example, in a theoretical scenario Figure 6.6, if a user under the AP1 request a content that has been previously downloaded by a user connected to a neighbouring AP, i.e. the AP3, they could start a D2D connection to share the content. The main goal is to see if there is a correlation between users connected to neighbours APs. In the previous sub-section we demonstrated that the more users connected more objects will be saved.

In our location, the Atrium of the Politecnico Figure 6.1, we have a distribution like Figure 6.7. As we said before, we couldn’t get the exact coverage area, so we assumed that only the one hop far away APs were neighbours, discarding the AP located on the other side of the Atrium:
We are going to start the analysis by studying the correlation in time for all the users for a whole hour and 15 minute intervals in the AP-CORTILE01, taking this AP as reference, we could show the progression in terms of objects saved as we consider more neighbouring AP. In the next figure we are going to take a look on the distribution of the users through the different AP:

From the Figure 6.9 we can see that the AP with more users is the AP-CORTILE05, because it is the router that is giving service to some administration offices like the International Office. It is also visible that the connection time of each user is not so long,
that is because we are analysing routers located in a passing zone, however, if we had looked at the routers located in the library we would have found longer connection times.

<table>
<thead>
<tr>
<th>AP</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORTILE 01</td>
<td>37</td>
</tr>
<tr>
<td>CORTILE 03</td>
<td>32</td>
</tr>
<tr>
<td>CORTILE 05</td>
<td>47</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>116</strong></td>
</tr>
</tbody>
</table>

**Figure 6.9:** Users connected to the neighbouring AP

### 6.2.1 AP-CORTILE01

In the table Figure 6.10 are shown the results for the AP-CORTILE01:

<table>
<thead>
<tr>
<th>Considering one hour interval</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of objects cached</td>
<td>1324</td>
<td></td>
</tr>
<tr>
<td>Total number of objects downloaded from the network</td>
<td>7400</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 6.10:** Results for the AP-CORTILE01

The differences between the performance in AP-DITAG01 and AP-CORTILE01 are remarkable, while in AP-DITAG01 the mechanism saved 40.458 objects in one hour, in the AP-CORTILE01 were 1.324. This behaviour is due to the difference in users in the different AP, because the D2D offloading technique depends directly on the number of users connected and the repetition of the downloaded content. Let’s see what happen if we consider the neighbouring AP.
6.2.2 AP-CORTILE01 + AP-CORTILE03

As it is shown in Figure 6.11, as we consider one more AP, and consequently more users, the number of objects cached is increasing, just with one more AP, this number is almost the double. It is also visible that both AP have their peak time on the interval 4.

6.2.3 AP-CORTILE01 + AP-CORTILE03 + AP-CORTILE05

Now that we are taking in count three different AP, with a total number of users of 116, we are getting remarkable results. This means that the D2D capacity to offload the net increase with the number of neighbouring users, this is very important indeed, because
it means that when offload the net is more needed the D2D technique shows a better performance.

To get a clear view of the behaviour of the three different cases, one single AP, two and three consecutive AP, in the different intervals, we build the plots Figure 6.14.
7 - Conclusions and Future Work

The next-generation mobile networks will have to cope with the increasing number of smartphones and consequently with the massive traffic increase in the cellular network. Recent studies have proposed D2D data-offloading mechanism as a solution to this problem. As we have explained at the section Related Work from chapter 3, some studies have addressed the technical design considerations that enable D2D communications. However, there are not many studies about the floating content. In this thesis, we have studied the downloaded content in a realistic scenario, the Politecnico di Torino, by analysing the content traces provided by the tool Tstat. We collected HTTP request and responses headers in the Polito WiFi APs, which provided the necessary information to characterize the content and to extract the potential savings that a D2D data-offloading mechanism could achieve.

We started by analysing the principal characteristics of the downloaded content in one hour, we proposed to definitions to study the popularity of the objects; as Number of Users or as Number of Requests. We conclude that the first definition is the most accurate one, because helps to reduce inflations, generated by the possibly wrong functioning of some users’ devices. We found that popularity decreases very fast, as a result, the number of hits is heavily unbalanced between the first objects and the rest of them. The popularity graphs follow the Pareto distribution. Moreover, we showed the popularity distribution of the objects in percentage, which represents the probability of an object to be hit each time a request is made, on a Wednesday from 12:00 to 13:00 the percentage for the most requested item is 0.00093%, thus we concluded that the most requested items are not so popular indeed. Next step was to analyse the size of the most requested objects, we conclude that is very difficult to find a correlation between the size and the popularity of the most requested objects, but we demonstrated that the majority of the content downloaded is in the range of 100B. We also build three different cases to analyse the popularity in function of the different size of the items provided by the server: The best-case, the average-case and the worst-case. The next step on the study was to analyse
the types of content downloaded by the users. To make D2D offloading technic possible, the user has to be able to cache it. So we divided the content between static and dynamic, focusing our efforts in the static that can be cached.

In the second part we studied the time correlation between the content and the users. We analysed three consecutive days, which showed a similar behaviour in terms of popularity, size and type of content. We demonstrated how popularity decreases, the differences between the two definitions of popularity, and how the second definition could lead to wrong results. We found that a bit less than 70% of the one-hundred most requested objects are static, and those are divided in 80% (text) and 20% (images), which is important for the performance of the D2D mechanism. Furthermore, we compared two consecutive days to prove that those similarities between consecutive days are due to the repetition of content among them, concluding that even in the worst-case scenario and taking into account 2,000 objects in the study, 55% of the objects were repeated, and that D2D would save 40.87% of the traffic. We also wanted to get those results for a less time interval, so we split the hour in 10 minute intervals and we conclude that the level of “repeatness” increase for a smaller set of objects as for a shorter time intervals. Moreover, we extracted the differences in terms of popularity between consecutive intervals, and we showed how the dispersion increases as the number of objects. As well we showed the differences between the three different size considerations, the best-case, average-case and worst-case, which saved 71.30%, 64.71% and 44.04 % of the traffic respectively. We concluded that besides those differences, the content downloaded has a great level of “repeativeness” and D2D have a great performance in data-offloading.

In the third part we analysed the popularity and the level of repetition of the objects but taking into account the spatial distribution of the users in the Politecnico. As the D2D data-offloading technique is a cooperative mechanism where users take advantage of the downloaded contents by their neighbours, it is important to analyse the content and the behaviour of the users in each AP. We encountered some constraints to proceed with the spatial analysis. We couldn’t get the corresponding content traces, so we had to match randomly users from the content traces and from the AP traces, to distribute randomly the content information among the different users connected. Moreover, we were not able to get the exact coverage of each AP, so we assume that just one-hop far away APs were neighbours. We started the spatial analysis by studying the users under one single AP, the AP-DITAG01. In one hour, the results were that D2D could save 40.458 objects. We also showed that the peak time was from 12:43 to 12:48, where the mechanism was able to save 18.741 objects. Next step on the spatial correlation analysis were to study the savings but taking into account the neighbouring APs. We started
Conclusions and Future Work

considering just two neighbours APs, the AP-CORTILE01 and the AP-CORTILE03. We showed that the AP-01 by himself, just could save 1.324 objects, due the low number of users connected to it. When we analysed the AP-01 and AP-03, and consequently we take into our study more users, the number of objects cached increase to 2.468. Finally, when we considered the three neighbouring APs, we got remarkable results, they could save 11.506 objects concluding that that D2D capacity to offload the net increase with the number of neighbouring users sharing content, that is very important indeed, because D2D technique shows a better performance when offloading the net is more needed.

In this paragraph, we are going to summarize the main directions for further work. A good first proposal would be to repeat the same study, but in other locations, someplace where users do not have anything in common, i.e. a crowded metro station, where the popularity of the content among the users would be very dispersed because users would be in a wide age range with different interests. This could be an opportunity to rebuild the study avoiding some constraints we had to assume in this thesis. Analyse the spatial correlation taking the corresponding content traces to get the most realistic results as possible. If this is not possible, our results suggest that next study should be based on more realistic mobility models; in this thesis we assigned random IPs to the users under the AP, we assumed that they were immobile, but in reality, users move along the different APs. Therefore, future studies in D2D performance modelling should include the effects of clustering as well as of correlation among users’ mobility patterns, and focus on performance over short time scales.

The next step would be to launch simulations to analyse special cases such: what happens when appears a viral content, what would happen if D2D mechanism was implemented in a more crowded location, such a sport or music events. It would be also interesting to analyse the evolution of the content spread and the availability duration of the floating content. The other proposal would be to study all the different ideas that could help to get a better performance of the D2D data-offloading mechanism, for example, the website providers could increase the static content on their websites. The network could push popular content to some devices periodically to spread it to the rest of the users; that could work for operating system updates, certificate revocation lists etc.

Remaining technical issues have to be addressed; including how to coexist with cellular users and how to deal with the interferences. Considering mobile terminal requirements, local cache size can be optimized as well energy impact has to be carefully evaluated too. Possible business strategies have to be studied too; there is a need to develop D2D applications which are attractive to both operators and users. D2D communications should
enable the operators to control their networks in order to provide better user experience and make profit accordingly. At the same time, it should be flexible and low-cost to compete with traditional D2D communications.

On the basis of the results of this research, it can be concluded that D2D mechanism is a promising solution to offload the cellular network, despite remaining technical issues and uncertain business strategies that have to be further investigated.
Bibliography


[8] 3GPP. Study on LTE device to device proximity services. RP-122009, December 2012.


