Smartphone traffic patterns

DANIEL CRESPO RAMÍREZ

KTH Information and
Communication Technology

Degree project in
Communication Systems
Second level, 30.0 HEC
Stockholm, Sweden

Smartphone traffic patterns

Daniel Crespo Ramírez

dacr@kth.se

Academic Supervisor & Examiner

Professor Gerald Q. Maguire Jr.

Industrial Supervisor

Klas Johansson

July 17, 2011

Stockholm, Sweden

Abstract

The growing popularity of new generation mobile terminals, known as ‘smartphones’, has increased the variety and number of such devices. These devices make use of the resources offered by Universal Mobile Telecommunication Services (UMTS) networks to access on-line services such as web browsing, e-mail, audio and video streaming, etc. UMTS networks have to deal with an increasing amount of data traffic generated by smartphones. Because of the fact that the smartphone is battery powered and is trying to satisfy the needs of both applications and human users there is a need to be smarter about how to manage both network and terminal resources.

This thesis explores the possibility of making a better use of the network and terminal resources by exploiting correlations in the events of the smartphone-generated traffic. We propose a mechanism, through which the network can predict if a terminal is going to produce data transmission or reception in a near future, based on past events in its traffic. According to this prediction, the network will be able to decide if it keeps or releases the resources allocated to the terminal. We analyze the benefits from the network and the terminal point of view. We also describe a method to estimate an upper bound of the time until the next transmission or reception of data in a near future.

We show that it is possible a reduction of the time that each terminal wastes in its maximum power consumption state, but this reduction implies a penalty in the transmission/reception throughput of the terminal. The reduction is not uniform for all terminals: terminals whose traffic presents a predictable behavior gain the most. Estimates of upper bounds of time until the next transmission or reception are more accurate if they are made taking as input information about interarrival times of previous packets.

Sammanfattning

Den växande populariteten för nya generationens mobila terminaler, så kallade "smartphones", har ökat både antal och sådana produkter. Dessa enheter utnyttjar de resurser som Universal Mobile Telecommunication Services (UMTS) att få tillgång till on-line tjänster såsom asweb webbläsning, e-post, ljud och video streaming, osv. UMTS-nät har hantera med en ökande mängd data som genereras trafik bysmartphones. På grund av det faktum att smartphone är batteridriven och försöker för att tillgodose behoven hos både applikationer och mänskliga användare det finns ett behov att vara smartare om hur man kan hantera både nätverk och terminaler resurser.

Den avhandling undersöker möjligheten att göra en bättre användning av nätverk och terminaler resurser genom att utnyttja samband i händelserna smartphone-genererade trafik. Vi föreslår en mekanism genom vilken nätet kan förutsäga om terminalen kommer att ta fram dataöverföring orreception i en nära framtid, baserat på tidigare händelser i trafiken. Enligt denna förutsägelse, kommer nätet att kunna avgöra om den håller eller frigör resurser till terminalen. Vi analyserar nytta nätet och terminalen synvinkel. Vi beskriver också en metod för att uppskatta övre gränsen för tiden till nästa sändning eller mottagning av data inom en snar framtidd.

Vi visar att det är möjligt att minska den tid som varje terminal avfall i sin maximal strömförbrukning staten, men denna minskning innebär en straffavgift överföring /mottagning genomströmning av terminalen. Minskningen är notuniform för alla terminaler där trafiken utgör en förutsägbart beteende vinna mest. Uppskattningar av övre gränserna för tid untilthe nästa sändning eller mottagning är mer exakta om de görs tar som indata information om interarrival gånger tidigare paket.

Acknowledgements

I would like to sincerely thank all the people who made possible this work. First of all, thanks to the people who made it possible directly: Professor Gerald Q. Maguire Jr., for his continuous and fast feedback; Klass Johansson, for his valuable advises and support; and Anders Näsman, for sharing a lot of technical knowledge about UMTS networks.

I would also like to thank all my friends in Stockholm. They have helped me to feel happy during this time, and without this feeling, this work would not have been possible.

I am very grateful to my family. Their unconditional support from the distance has been essential. A part of this work is theirs.

I would like to mention my friends in Spain. Their support has been very important. For me, they are the best people in the world. And last, but not least, a special mention to Patricia, for her infinite love and comprehension during this year.

Table of Contents

List of Figures

List of Abbreviations and Acronyms

3GPP	3rd Generation Partnership Project
BCH	Broadcast Channel
cdf	Cumulative distribution function
CPC	Continuous Packet Connectivity
CPCH	Common Packet Channel
DCH	Dedicated Channel
DPCCH	Dedicated Physical Control Channel
DRX	Discontinuous Reception
DSCH	Downlink Shared Channel
DSL	Digital Subscriber Loop
DTX	Discontinuous Transmission
E-DCH	Enhanced DCH
FACH	Forward Access Channel
HSDPA	High Speed Downlink Packet Acess
HS-DSCH	High Speed Downlink Shared Channel
HSPA	High Speed Packet Access
HS-PDSCH	High Speed Physical Downlink Shared Channel
HS-SCCH	High Speed Shared Control Channel
HSUPA	High Speed Uplink Packet Access
IAT	Interarrival time
OS	Operating System
PCH	Paging Channel
pdf	Probability density function
QoS	Quality of Service
RACH	Random Acess Channel
RNC	Radio Network Controller
RRC	Radio Resource Control
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
URA	UTRAN Registration Area
UTRA	Universal Terrestrial Radio Access
UTRAN	UMTS Terrestrial Radio Access Network
WCDMA	Wideband Code Division Multiple Access

1. Introduction

Mobile devices with advanced operating systems have become extremely popular recently. Advanced operating systems are defined as those which are able to run independent applications. Some of the most well known operating systems for mobile devices are Google’s Android, Nokia’s Symbian OS, Microsoft’s Windows Mobile, or Apple’s iOS. Apple's operating system for both their iPhone and iPad is iOS. Mobile phones running one of these operating systems are generally called ‘smartphones’.

These mobile devices have become popular because, besides having the functions of a traditional mobile phone (voice calls, short messaging service –SMS), they also offer a wide variety of applications which allow the user to access on-line services such as web browsing, e-mail, audio and video streaming, etc. These applications often require high speed connections to remote servers through wide area wireless communication infrastructures. Nowadays, Universal Mobile Telecommunication Services (UMTS) networks, also known as 3G mobile networks, are the most widespread wide area communication infrastructures.

Smartphones place high requirements on low battery consumption. As battery power is a limited resource, the consumption of power has to be very well managed in order to increase the standby and operating times of these devices.

UMTS networks must deal with the increasing amounts of data traffic being generated by smartphones. As a result, there is a need to manage both network and device resources in a smarter way than ever before. Earlier the load due to data traffic in mobile networks was significantly lower and few applications other than voice calls and SMS were used. In addition, while there are some users who talk a lot on their phones, there are many more users who expect to be able to use web browsing and other applications for a large part of their day.

1.1. Problem definition

One of the key challenges of 3G mobile networks today is to provide connectivity to an increasing number of smartphones, everywhere, and with the best possible service. To achieve this, effective management of network resources is important. For example, this means adapting the resource management algorithms to the particularities of the data traffic generated by smartphones.

In order to adapt resource management to user behavior, a first step is to learn about the characteristics of smartphone-generated traffic. In the case of 3G terminals all of the traffic is packet based traffic. Studying this traffic may allow us to identify ‘patterns’. By ‘pattern’ we mean a ‘regular, discernible sequence of events repeated in time’. An ‘event’ in this context is the sending or reception of a packet. If it is possible to identify patterns in the traffic and find correlations between events, resource management algorithms could exploit knowledge of these patterns and correlations on both the network and terminal sides. Technically, in UMTS networks, the ‘network side’ refers to the UMTS Terrestrial Radio Access Network (UTRAN), and ‘terminal side’ refers to the User Equipment (UE), in the case of this thesis this will be a smartphone.

1.1.1. Network controlled mechanisms

From the UTRAN, the key aspects of the resource management governed by the Radio Network Controller (RNC) are:

- Channel scheduling, and
- Continuous Packet Connectivity (controlling Discontinuous Transmission and Reception, or DTX/DRX).

1.1.1.1. Channel scheduling

Channel scheduling defines what network resources are allocated to each terminal for what period of time. This scheduling could be done more efficiently if the network can ‘predict’ the next events in terms of what traffic each UE will generate or receive. These ‘predictions’ (based on the identified patterns) must be made with a certain accuracy.

Channel scheduling affects the battery consumption of the UEs, since power consumption due to the radio subsystem is directly related to the management of the Radio Resource Control (RRC) states in the terminals. The RRC state of a UE defines its level of connectivity. The higher the connectivity level the UE has, the higher the power consumption will be. Further details of this will be given in section 2.1.3.

1.1.1.2. Discontinuous Transmission and Reception

Continuous Packet Connectivity (CPC) is a feature of 3G mobile networks defined in Release 7 of UMTS specifications. CPC offers a set of features which avoid the drawbacks due to the High Speed Packet Access (HSPA) feature introduced in Release 5. Further details of CPC will be given in section 2.1.5.

Discontinuous transmission and Reception (DTX/DRX) are part of CPC. They allow UEs to transmit and receive the control information related to HSPA transport channels in a discontinuous way when there is no user data traffic. This reduces the need for the terminal and the network to transmit this information, which in turn reduces cell interference, hence increasing the network’s capacity. DTX/DRX also reduces the power consumption of the UE, as it does not need to transmit or receive at all times.

To implement DTX/DRX, CPC defines a number of parameters: inactivity timers, transmission and reception cycles, etc. If we know and can accurately recognize traffic patterns, then it is possible to choose suitable values for each of those parameters. This requires that the network recognizes a known traffic pattern is ‘happening’, thus enabling it to dynamically adjust the parameters in a more appropriate way for this pattern.

1.1.2. User equipment (UE) controlled mechanisms

From the UE’s point of view, a better use of the resources can be achieved by exploiting the knowledge of traffic patterns. Terminals in an ‘idle’ state must transition to a ‘connected’ state in order to send or receive data, and then return to an ‘idle’ state. The decision of when to connect and disconnect could be made by the applications in the UE while taking into account information about known traffic patterns and the current requirements of the applications. Correlations between sending and receiving events can help UEs to decide when is the best moment to connect and disconnect in order to reduce their battery consumption and/or reduce their contribution to network load. Different strategies will be defined and evaluated for different applications.

1.1.3. Research questions

Some 'high level' research questions motivated by the above are:

- How can traffic patterns be characterized and recognized?
- What information can be extracted from the presence of patterns in the traffic?
- How can the channel scheduling strategy and the DTX/DRX be adapted to the traffic patterns in practical terms?
- How much can we improve the network throughput by knowing traffic patterns?
- How large are the potential battery savings we can achieve?

1.2. Research approach

In order to identify and characterize patterns in the data traffic generated by smartphone terminals in 3G mobile networks, we have utilized logs of user data packets captured at the Gi interface of a UMTS network (see section 2.1.1 for a discussion of this interface). Then, we define an abstraction called a 'packet burst' (described in section 3.1). This abstraction will allow us to describe the flow of events in the traffic of each user more clearly.

Once the events are identified, we explore the possibility of predicting the next event in the traffic of a specific user, based on past events in this user's traffic. Since there are strong correlations in these events, we will hopefully be able to make predictions with an acceptable accuracy. Section 3.3.2 discusses the effect of differences in accuracy.

We also analyze a possible method to estimate an upper bound for the time between the arrival of a packet and the arrival of the next packet.

The next step is to analyze the benefits and drawbacks based upon these predictions in terms of each different mechanism, such as the channel sscheduling and DTX/DRX at the UTRAN, and the decision of when to connect at the UE. This analysis is made in terms of the resources used within the UTRAN and the observed battery consumption of the terminals.

One complication is that the actual traffic a UE generates in a network is strongly related with network conditions. For instance, consider a user who wants to browse his or her favorite web pages via a smartphone. If this user is in a location that has poor connectivity (due to low signal level), or if the network is overloaded, then the user will experience long delays. This user may give up after attempting to browse one or more pages. However, if the network conditions are favorable, this same user might browse 10 different pages, generating more *user* network traffic. As a result, when we study network traffic we must evaluate the 'change' that can occur under different network conditions, in order to understand how this change in network conditions will cause a change in the *user* traffic in the network. This leads us to an iterative process of traffic study and network improvement.

1.3. Thesis outline

Chapter 1 presents an introduction to the problem and a set of questions to be answered at the end of this work. Chapter 2 provides the necessary technical background in order to understand the work. We explain technical concepts about UMTS network and about models of packet traffic. In chapter 3 we explain the methods we used to achieve our goals. Chapter 4 presents the results of applying the methods explained in chapter 3 to the traffic of a real UMTS network. This thesis ends with chapter 5, in which we give the answers to the questions presented in section 1.1.3, along with suggestions about the further work to be done in this area.

2. Background

This chapter presents an introduction to the basic concepts needed to understand this thesis. The chapter will focus on three topic: UMTS networks, WCDMA technology, and packet traffic models.

Section 2.1 provides background on WCDMA technology and how UMTS networks work. Most of the material is based on [1] and [2]. Section 2.2 discusses the models used and how to characterize the properties of the packet-based traffic. This material is primarily based on [4].

2.1. Wideband Code Division Multiple Access

Wideband Code Division Multiple Access (WCDMA) technology has been adopted as one of the standard air interfaces for the mobile networks known as Universal Mobile Telecommunication Services (UMTS) networks. WCDMA was specified by the 3rd Generation Partnership Project (3GPP), a joint standardization project of standardization bodies from Europe, Japan, Korea, USA, and China. 3GPP refers to WCDMA as Universal Terrestrial Radio Access (UTRA).

The outline of this section is the following. The structure of a UMTS network is described in subsection 2.1.1. The transport channels defined in the WCDMA specifications are described in subsection 2.1.2, including an introduction to High Speed Packet Access (HSPA). In subsection 2.1.3 the Radio Resource Control (RRC) protocol is presented, with descriptions of the operational modes defined for the terminals. Some notes about packet scheduling are presented in section 2.1.4, including a description of the ‘Fast dormancy’ feature. Finally, subsection 2.1.5 presents a description of the Continuous Packet Connectivity (CPC) feature.

2.1.1. Structure of a UMTS network

Figure 1 shows an overview of a UMTS network. A brief description of its elements is given below.

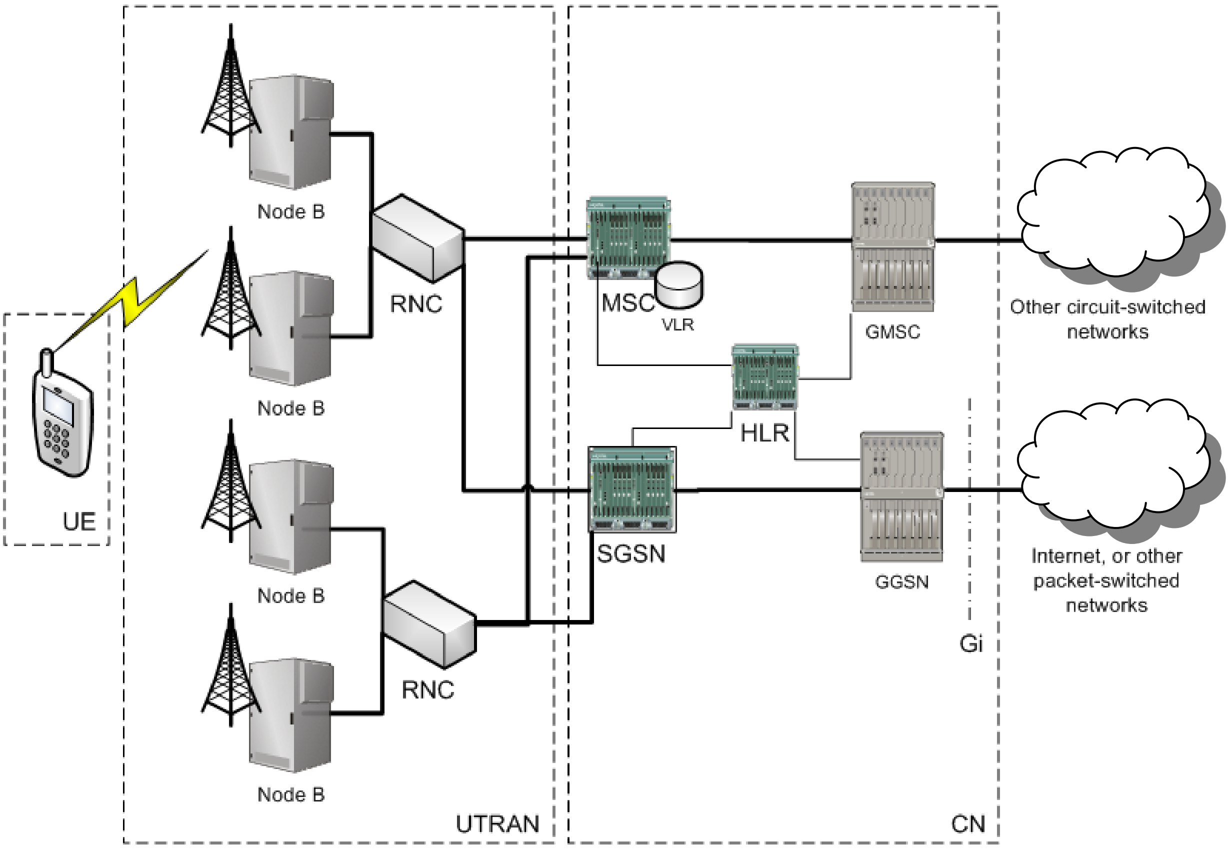

Node B
Node B
Node B
Node B
RNC
RNC
UE
MSC
VLR
GMSC
HLR
SGSN
GGSN
Gi
Other circuit-switched networks
Internet, or other packet-switched networks
UTRAN
CN

Figure 1: Overview of a UMTS network

We can see there are three groups of network elements:

Core network (CN)	Provides switching, routing, and transit for user traffic
UMTS Terrestrial Radio Access Network (UTRAN)	Handles all the radio related functionality. It provides the air interface access method for UEs
User Equipment (UE)	Interfaces the mobile equipment via the radio network to external networks.

Figure 1 also shows the Gi interface, which is the point where the UMTS network communicates with other external packet-switched networks, such as the Internet.

2.1.1.1. User Equipment

The user equipment (UE) consists of two elements:

- Mobile Equipment (ME) is the radio terminal used for communication.
- UMTS Subscriber Identity Module (USIM) is a smart chip card that contains the subscriber’s identity, authentication and encryption keys, and subscription information.

2.1.1.2. UMTS Terrestrial Radio Access Network (UTRAN)

The UTRAN consists of two different elements:

- Node B, also known by the more generic term ‘base station’, moves data between the two interfaces Uu (air interface) and Iub (interface with Radio Network Controller). It performs channel coding and interleaving, rate adaptation, and spreading. The geographic area to which a Node B provides service is called a *cell*.
- Radio Network Controller (RNC) manages the radio resources in its domain (set of Node Bs connected to it). For a given UE, its Serving RNC (SRNC) is the RNC to which the terminal communicates both user and signaling data. There can be other RNCs which control cells used by the UE, these are called the UE’s Drift RNCs.

2.1.1.3. Core Network

The core network provides the infrastructure which connects all of the RNCs between them and to other networks. The core network also provides the infrastructure for mobility, authentication and authorization, accounting, billing, etc. It consists of the following major subsystems:

- The Home Location Register (HLR) is a database which stores master copies of subscribers’ service profiles and macro-location information (indicating the subscriber’s current MSC/VLR).
- The Mobile Services Switching Centre (MSC) performs circuit switching operations.
- The Gateway MSC (GMSC) is a switch that connects the UMTS network with an external network.
- The Visitor Location Register (VLR) is a database of user profiles with more precise information about the location of UEs within the network. Each MSC has an associated VLR.
- The Serving General Packet Radio Service Support Node (SGSN) routes packets through the network to support packet switched services.
- The Gateway GPRS Support Node (GGSN) is a SGSN which connects the core network with an external packet switched network such as the internet, over the Gi interface.

2.1.2. Transport channels in WCDMA

Communication between UEs and RNCs is structured into a set of protocol layers. There are separate protocol stacks for the user information and for the control information. Every protocol stack has a ‘transport network layer’, in which different types of transport channels are defined. Details of this are given below.

Data generated at higher layers (both user data and control data) are carried over the link between UEs and RNCs via transport channels. These transport channels are mapped at the physical layer to physical channels. A transport channel is said to be a downlink channel if the information sent through it goes from the RNC to the UE. If the information goes from the UE to the RNC, then the channel is said to be an uplink channel.

Release 99 of UMTS specifications specifies two main types of transport channels: dedicated channels (DCH) and common channels. Later, Release 5 of the UMTS specifications introduced High Speed Packet Access (HSPA); along with definitions of new types of channels and scheduling techniques.

2.1.2.1. Release 99 transport channels

As noted above, there are two types of transport channels defined in Release 99: dedicated channels (DCH) and common channels. Common channels are network resources shared by all UEs within a single cell. There are six different common transport channel types defined: Random Access Channel (RACH), Forward Access Channel (FACH), Paging Channel (PCH), Broadcast Channel (BCH), Uplink Common Packet Channel (CPCH), and Downlink Shared Channel (DSCH). The common transport channels required for basic network operation are RACH, FACH, and PCH; while the others are optional. A description of each kind of transport channel is given below:

Dedicated Transport Channel (DCH)

Carries all the information intended for a specific single user. This information can be user data (such as speech frames) or control information (handover commands or measurement reports from the UE). A DCH resource is identified by a certain code within a specific frequency band. Communication is bidirectional.

Random Access Channel (RACH)

This uplink common transport channel carries control information from a UE to the RNC (such as a request for a dedicated connection).

Forward Access Channel (FACH)

This downlink common channel carries control information for all the UEs located within a cell. A cell can have more than one FACH, and at least one of them contains low bit-rate data to be received by all UEs in the cell.

Paging Channel (PCH)

This downlink common channel carries data for the paging procedure. The paging procedure is executed when the network wants to establish communication with a certain UE, for instance, in case of an incoming call to this UE. A paging message is sent through the PCH of all the cells within the paging area where the UE is expected to be. The design of the PCH affects the power consumption of the UEs: the less frequently the UE has to listen to the PCH for possible incoming pages, the lower the power consumption will be, but the higher the delay in responding to a page.

Broadcast Channel (BCH)

This is a common transport channel used to transmit specific information about the network or a given cell, such as the available random access codes and access slots.

Uplink Common Packet Channel (CPCH)

This is an extension of the RACH intended to transmit packet-based user data. The reciprocal downlink channel to CPCH is the FACH.

Downlink Shared Channel (DSCH)

This is a common channel intended to transmit dedicated user data and/or control information, but it can be shared by several UEs.

The transport channels defined in Release 99 terminate at the RNC and the retransmission procedures are located in the RNC. This implies that the presence of Node Bs is transparent to these transport channels.

2.1.2.2. High speed packet access (HSPA)

HSPA was introduced in Release 5 of WCDMA specifications to increase transmission and reception bit rates. This is achieved by introducing additional intelligence in Node Bs to perform retransmissions and transmission combining. Giving each Node B control over its own retransmissions leads to faster retransmissions and lower latencies.

New transport channel types were defined to carry user data, specifically the High-Speed Downlink-Shared Channel (HS-DSCH) for downlink traffic and the Enhanced Dedicated Channel (E-DCH) for uplink traffic. These channels provide support for higher bit rates. Release 5 specifies bit rates of up to 10.8 Mbps on the downlink and up to 5.7 Mbps over the uplink.

HSPA introduces a new scheduling schema. The HS-DSCH is dynamically allocated to a specific user for a short period of time, during which the user has most of the cell’s capacity. This is done when conditions are favorable for this UE. Every 2 ms, the allocation of the high data rate channel can be changed to another user.

New physical channels are also defined to carry the HS-DSCH and all the control information related to it, specifically these are: the High Speed Physical Downlink Shared Channel (HS-PDSCH), the Dedicated Physical Control Channel (DPCCH), and the High Speed Shared Control Channel (HS-SCCH). The HS-PDSCH carries the user data transmitted through the HS-DSCH. The higher bit rate is achieved using 16 QAM modulation (in addition to the Release 99 QPSK modulation) and new redundancy strategies in the level of channel coding. The DPCCH is an uplink dedicated channel carrying control information from a specific UE to the Node B, such as acknowledgements of packets received on the HS-PDSCH and Channel Quality Indication (CQI) reports. The High-Speed Shared Control Channel (HS-SCCH) carries the key timing and coding information for HS-DSCH demodulation.

2.1.3. Radio Resource Control (RRC) Protocol

The RRC protocol generates most of the signaling traffic between the UE and the UTRAN. RRC messages allow the set up, modification, and release of resources in the RNC. The RRC protocol defines the basic operational modes and states of the UEs. The state a UE is in limits the channels it can use.

Figure 2 shows the operation modes and states defined in the RRC protocol and the possible transitions between them.

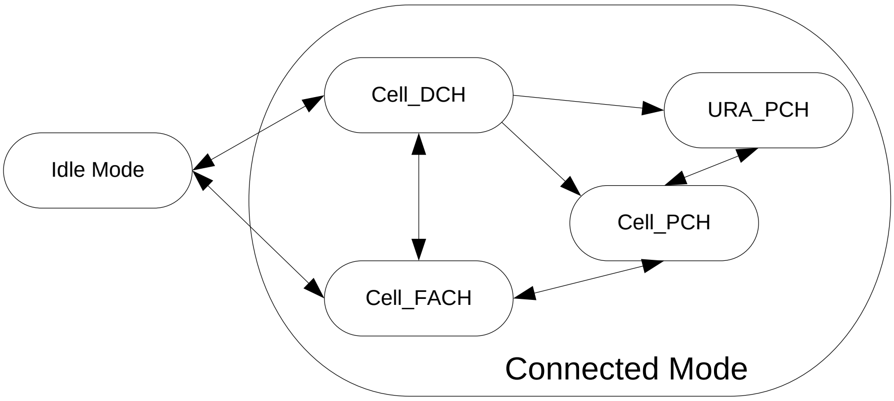


Figure 2: UE modes and RRC states in connected mode

2.1.3.1. Idle mode

When the UE is switched on, it selects a public land mobile network (PLMN) to connect to, chooses a suitable cell of this network and tunes to the control channel. The UE remains in idle mode. In this mode, the UE is identified by the international mobile subscriber identity (IMSI), the temporary mobile subscriber identity (TIMSI), and the packet TIMSI (P-TIMSI); the first identity is provided by a USIM card in the UE and the later two identities are assigned by the network after the UE authenticates itself to the network. The UTRAN knows which paging area this UE is in, but other than this the UTRAN has no information about individual UEs in idle mode, so it cannot address them individually, but communicates through transmissions to all UEs in cell. RNCs can only address traffic to specific UEs if they are in Cell_DCH or Cell_FACH state. The UE remains in idle mode until it transmits a request to establish an RRC connection.

2.1.3.2. Connected mode: Cell_DCH

In Cell_DCH state, a physical channel is allocated to the UE. The serving RNC (SRNC) of the UE knows which cell this UE is in. The UE sends measurement reports according to the measurement control information it receives from the SRNC. The UE can monitor the Downlink Shared Channel (DSCH) and the Forward Access Channel (FACH) in this state. Since communication via DCH is bidirectional, both the transmitter and receiver of the UE must be active, so the power consumption in this state is high.

2.1.3.3. Connected mode: Cell_FACH

In Cell_FACH state, no dedicated physical channel is allocated to the UE, rather the UE uses FACH and RACH channels to communicate signaling messages and small amounts of user data. The UE can also listen to the Broadcast Channel (BCH) of the cell, in order to receive system information, and can use uplink common packet channels (CPCH). The RNC knows the location of the UE on a cell level. If the UE performs a cell reselection, it will send a Cell Update message.

2.1.3.4. Connected mode: Cell_PCH

In Cell_PCH, the location of the UE is known by its SRNC on a cell level, but this UE can only be reached through the paging channel (PCH). The power consumption is lower, because monitoring of the PCH utilizes discontinuous reception functionality. If the UE needs to perform a cell reselection, it transitions to the Cell_FACH state to perform the Cell Update procedure and then returns to Cell_PCH if no other activity is triggered.

2.1.3.5. Connected mode: URA_PCH

A UTRAN Registration Area (URA) is an area covered by a number of cells. The URA_PCH state is similar to the Cell_PCH state. The difference is that the UE does not perform a Cell Update procedure after cell reselection. Instead, the UE learns the URA identity from the BCH, then if the URA identity has changed after a cell reselection, it performs a URA update procedure to notify the SRNC of its new location. To perform the URA update, the UE transitions to Cell_FACH state, and when done it will return to the URA_PCH state. A cell can belong to one or more URAs. Only when the UE cannot find its latest URA identification in the list of URAs of a new cell will it execute the URA Update procedure. After the URA update, the location of the UE will be known by its RNC on a URA level. Since the URA covers a larger area than a cell, URA Updates will be less frequent than cell Updates and the power consumption will be even lower than when in Cell_PCH state. However, the cost (in network resources) of paging this UE increases as it has to be paged in all of the cells of this URA.

The relative order of these states from highest to lowest power consumption is: Cell_DCH, Cell_FACH, Cell_PCH, URA_PCH, and Idle.

2.1.4. Packet scheduling

Scheduling controls the allocation of the shared resources among users for the period of time for which the scheduled is prepared. ‘Packet scheduling’ provides support to packet switched services (such as messaging, email, web browsing, streaming, etc.). There are two aspects of this packet scheduling: the control of the utilization of the RRC states by each user, known as ‘User-Specific Packet Scheduling’, and the control of the sharing of the radio resources between simultaneous users, known as ‘Cell-Specific Packet Scheduling’.

2.1.4.1. User-Specific Packet Scheduling

This part of packet scheduling manages the transitions between the RRC states of the UEs. This involves the allocation of channels and the adjustment of the bit rates according to the radio power limitations and the capacity of the network.

Since all UEs in a UMTS network operate in the same frequency band, the received power per bit at the base station should be equalized for all UEs, in order to avoid devices which are near the base station overpowering those which are far from it. RNCs have mechanisms to tell UEs how much power they are allowed to transmit with. A higher bit rate transmission implies more power emission for a given distance between transmitter and receiver.

The UE has an uplink buffer which stores traffic to be sent to the RNC. There should be a traffic volume threshold such that if the traffic in the buffer does not exceed this threshold, then this traffic is sent through the RACH while the UE is in the Cell_FACH state. If this threshold is exceeded, then a DCH is allocated at the minimum bit rate and the UE enters the Cell_DCH state. Before increasing its bit rate, the link power and the capacity of the base station must be checked; if there are no restrictions, then the bit rate of the DCH can be increased.

After data transmission, inactivity timers control the transitions from Cell_DCH to lower power consumption states. After some inactivity time the DCH it is released to avoid unnecessary waste of network resources, and the UE transitions to Cell_FACH state, during which it is still able to transmit through the shared FACH channel. Once again, after some inactivity time, the UE transitions to a lower power consumption state, such as one of the paging states (Cell_PCH or URA_PCH) or Idle.

Fast dormancy

The mechanism of inactivity timers controlling transitions to lower power consumption states described above is inefficient: if the UE is inactive, then a lot of energy is wasted between the moment the data transmission ends and the moment the UE enters a lower power consumption state. Therefore, if the UE knows that once it ends a transmission via DCH it will not transmit again for a period of time, then it makes no sense to keep the UE in Cell_DCH for a time and then transition to Cell_FACH for yet another time; instead the UE can transition directly to the Idle state.

This is the underlying idea behind the introduction of the fast dormancy feature. An initial version of it allowed the UE to trigger a release of the connection and transition to Idle mode after a transmission, by sending a RRC protocol indication message. Newer versions of fast dormancy specify that is the network which decides when to trigger the release of the connection after receiving a notification from the UE, and that the network decides whether to ask the UE to transition to the Idle or to a paging state.

Fast dormancy allows significant battery savings. Even greater advantage can be taken by moving the UE to a paging state (Cell_PCH or URA_PCH) rather than to the Idle, since the cost of going from a paging state to Cell_DCH or Cell_FACH again is lower than going from the Idle state, while there is not a significant difference in the battery saving between these low power states (due to the relatively low duty cycle of listening for pages).

2.1.4.2. Cell-Specific Packet Scheduling

The cell-specific packet scheduler divides the non-real time capacity of the cell between simultaneous users. The non-real time capacity is the available capacity of the cell for low priority traffic.

There are four classes of packet traffic defined according to its quality of service (QoS) requirements. They are, in decreasing order of latency requirements: conversational, streaming, interactive, and background. Conversational traffic and part of the streaming traffic, due to their strict requirements for low latency, are only transmitted through DCHs, thus they have a guaranteed (minimum) bit rate. The DCHs will be directly allocated for the UEs transmitting this kind of traffic. The part of the cell capacity used by these connections is called the ‘real time capacity’ of the cell. The remaining part of the cell capacity is the non-real time capacity. The cell-specific packet scheduler will manage this part of the cell capacity, dividing it between UEs whose traffic has weaker requirements of latency, and establishing priorities between UEs.

The cell-specific packet scheduler operates periodically. It takes as input for its task the following information:

- Total Node B estimated power,
- Capacity used by non-real time bearers,
- Target load level from network planning parameters, and
- Bit rate upgrade requests from the user-specific packet scheduler.

If the load is less than the target load, then higher bit rates can be allocated. However, interference levels must be maintained within planned values.

QoS parameters of the non-real time traffic are also taken into account. Higher priority bearers are allocated before lower priority bearers.

2.1.4.3. Packet Scheduling in HSPA

As was mentioned earlier, introduction of HSPA requires introduction of additional intelligence in Node Bs. Because HS-DSCH is a shared channel, scheduling is crucial to achieving good performance. The available resources must be distributed among users in a fair and efficient way.

The scheduling to be done in the Node Bs is not defined in the specifications. However, an evaluation of some possible scheduling algorithms is presented by Janevski and Jakimoski [3]. These algorithms are:

Round robin
Users are served in a cyclic order. The scheduler allocates the resources to the user who has not been served for the longest time.

Maximum C/I
The scheduler allocates the resources to the

	user with the best instantaneous channel quality, in terms of the carrier-to-interference (C/I) ratio.
Fair Channel-Dependent Scheduler	A hybrid of the previous two algorithms.

2.1.5. Continuous Packet Connectivity (CPC)

User-generated packet-data traffic carried over HSPA is often bursty as there are activity periods, when packets are sent and received, followed by inactivity periods, when no information is sent or received. Releases 5 and 6 of UMTS specifications specified that, during the periods of user data inactivity, E-DCH and HS-DSCH channels are kept configured to be able to transmit user data, in order to make the latency experienced by the user as low as possible.

Keeping the channels configured comes at a high cost, both from the network and from the UE points of view. From the network’s side, it leads to a high degree of uplink interference in cells and a high work load at RNCs, because each UE needs to transmit continuously on the DPCCH channel, and the control information has to be continuously processed. From the UE’s point of view, keeping the channels configured has a negative impact on battery consumption as even though there is no user data transmission or reception, the UE needs to have its transmission and reception circuitry switched on, in order to continuously transmit control information via DPCCH and continuously monitor the HS-SCCH for incoming control information from the RNC.

To avoid these drawbacks due to HSPA, a set of additional features were introduced in Release 7 of the UMTS specifications. This set of new features is known as ‘Continuous Packet Connectivity (CPC)’. The new features are Discontinuous Transmission (DTX), Discontinuous Reception (DRX), and HS-SCCH-less operation mode. Each of these will be described in more detail below.

2.1.5.1. Discontinuous Transmission (DTX)

A naïve approach to solve the described drawbacks of HSPA would be to not transmit any control information through the DPCCH when there is no user data transmission. The UE could conserve battery power by switching off its transmission circuitry, and the cell interference would be reduced, so the rest of the UEs could transmit at lower power; however, this would make it difficult to maintain uplink synchronization, producing high delays when a new burst of user data needs to be transmitted. It would also have a negative impact on power control, as the UE would not get feedback from the network about the power it is allowed to transmit. Thus, occasional DPCCH activity needs to be sent.

Uplink Discontinuous Transmission (DTX) allows the UE to automatically stop the continuous DPCCH transmission when there is no user data transmission in the E-DCH channel. In this situation, the UE will periodically transmit a DPCCH burst according to a UE specific DTX cycle, configured in the UE and the Node B by the RNC. Two cycles are defined, a ‘short’ one (cycle 1) and a ‘long’ one (cycle 2), which is an integer multiple of the short cycle. When the UE enters DTX mode, it will send periodical DPCCH bursts according to cycle 1; after some configurable inactivity time, it will only send a DPCCH burst according to cycle 2. Since cycle 2 is longer, the transmission of the bursts will be less frequent. When in DTX mode, the UE is not allowed to send user data through the E-DCH until a DPCCH burst starts. While user information is being sent, transmission of DPCCH will be continuous; after a burst of user data, the UE will enter DTX mode again using cycle 1.

Since the transmission of the control information is discontinuous, synchronization between network and UE becomes important. The DTX mode can start some time after the end of the E-DCH transmission, to facilitate synchronization. Also, when there is no DPCCH transmission, the UE cannot get any power control feedback from the Node B. 'Preambles' and 'postambles' are sent before and after, respectively, the DPCCH bursts, for power control purposes. UE-specific time offsets can be set in order to spread the DPCCH transmission occasions from different UEs in time.

The relevant parameters governing DTX are the following.

UE_DTX_Cycle_1 Defines the time between bursts of DPCCH activity when the UE first enters the DTX mode.

UE_DPCCH_Burst_1 Length of DPCCH bursts in cycle 1.

Inactivity_Threshold Time of inactivity in the E-DCH, after the UE enters DTX mode in cycle 1 and until the UE changes from cycle 1 to cycle 2.

UE_DTX_Cycle_2 Defines the time between bursts of DPCCH activity after a certain inactivity time since the UE entered the DTX mode in cycle 1. UE_DTX_Cycle_2 = n * UE_DTX_Cycle_1, where n is a positive integer.

UE_DPCCH_Burst_2 Length of the DPCCH bursts in cycle 2.

Enable_Delay Time between the end of an E-DCH burst and the moment that DTX mode actually starts. During this time, DPCCH transmission will still continuous.

Figure 3 illustrates the different transmission states of the DPCCH.

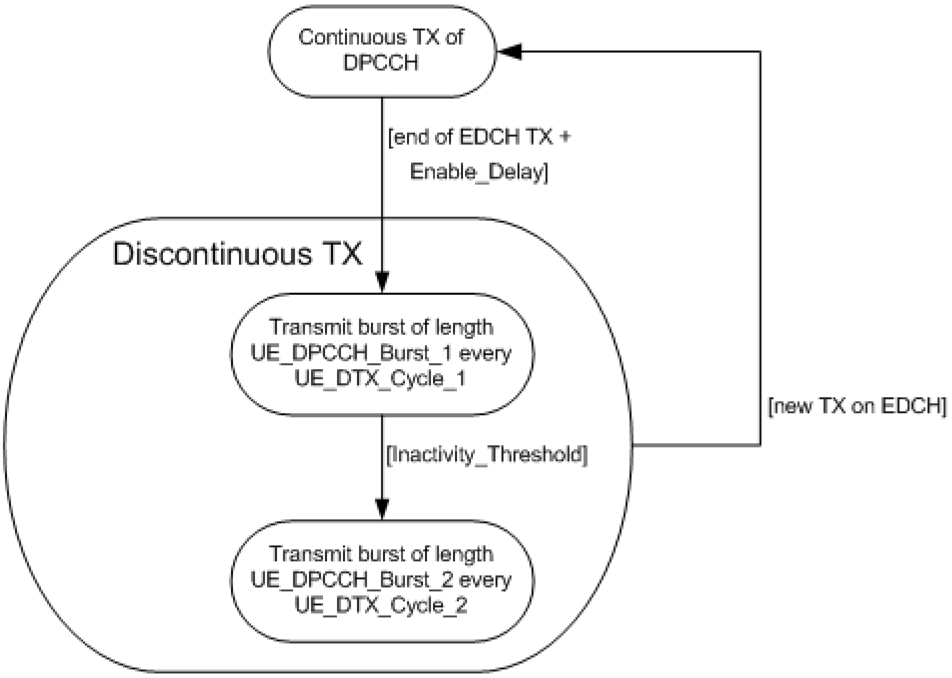

Continuous TX of DPCCH
[end of EDCH TX + Enable_Delay]
Discontinuous TX
Transmit burst of length UE_DPCCH_Burst_1 every UE_DTX_Cycle_1
[Inactivity_Threshold]
Transmit burst of length UE_DPCCH_Burst_2 every UE_DTX_Cycle_2
[new TX on EDCH]

Figure 3: State diagram of the DPCCH

Uplink DTX reduces uplink interference. Also, it allows discontinuous reception in Node Bs, which is useful to save processing resources as the received signal from UEs is not continuously processed. These factors increase the cell's capacity.

2.1.5.2. Discontinuous Reception (DRX)

CPC also introduces downlink discontinuous reception (DRX), to be used in combination with the previously described DTX feature.

The UE is required to monitor the downlink control channel HS-SCCH. DRX allows the network to limit when the UE must monitor the channel to check if downlink user data transmission is starting again. The rest of the time, the UE can switch off its receiver. A UE DRX cycle starts after a certain period of inactivity of the HS-DSCH channel. DRX cycles must match DTX cycles because the UE needs to receive power control commands from the Node B in all downlink slots corresponding to 'active' uplink slots (i.e., slots where the UE transmits).

2.1.5.3. HS-SCCH-less operation mode

High Speed Shared Control Channel (HS-SCCH) is a downlink channel used to carry downlink signaling related to HS-DSCH transmission. It provides the necessary timing and coding information to the UE to enable it to listen to HS-DSCH and decode the data intended for this UE.

The last feature of CPC is the HS-SCCH-less operation mode. When this mode is enabled, no control information about the HS-SDCH is sent through the HS-SCCH. Instead, the UE has to blindly decode the transport format used on the HS-DSCH from among a set of predefined formats. Currently the number of possible formats is limited to four. As a result the receiver decodes the received signal in each of the four ways until it finds a decoded version that makes sense.

2.2. Internet traffic models

The amount of data being carried over packet switched networks, specially the Internet, has grown exponentially during the last decades. This growth has motivated network operators to properly dimension their networks. To ‘dimension a network’ means estimating the required capacity of its nodes and links such that they are able to carry the actual amount of traffic that the link experiences, while optimizing expenses. This optimizing of expenses implies the need to cleverly manage resources, in order to avoid unnecessary waste of network capacity and its consequent potential waste of economic resources.

Mechanisms to properly dimension packet switched networks are needed, in order to optimize expenses as mentioned above. To propose a suitable mechanism, it is desirable to evaluate the response of the network to a given set of circumstances, and this is frequently made through simulations. A simulation model is a representation of the key elements of the network. A ‘traffic model’ is a stochastic process which represents the actual traffic measured in a network in a simulation model. Traffic models are used to predict the behavior of actual traffic streams, so ideally they should preserve all the statistical properties of the original traffic. There are some desirable properties for a traffic model, such as: it should be defined by a small number of parameters, its first and second order statistics should match those of the actual measured traffic, and if the traffic were fed through the model the results should accurately predict those of the real traffic stream fed into an actual network.

The dimensioning methods for circuit switched networks, based on the Erlang model for telephony, do not work in packet switched networks. This occurs because the Erlang model does not fit the properties of packet traffic. There have been many research efforts to find a traffic model which fits the properties and particularities of packet switched networks

Measurements of packet traffic have shown that this traffic exhibits long-range dependence, self similarity, and heavy-tailed distribution of interarrival times [6]. These properties are associated with the autocorrelation of stochastic processes and with probability distributions. The relevant mathematical concepts are introduced in the next subsections.

2.2.1. Definition of Autocorrelation

The autocorrelation function of a signal is the cross-correlation of the signal with itself. Roughly speaking, it is a measure of the similarity of the signal with time shifted versions of itself. Given a discrete-time real function X_t, its autocorrelation function $R(k)$ is defined as:

$$R_{XX}(k) = \sum_{t=-\infty}^{\infty} X_t \cdot X_{t-k} \quad , \qquad k = 0, \pm 1, \pm 2, \ldots$$

2.2.2. Properties of packet traffic

In many simulation models, the sources of the network traffic are stochastic processes. For the model to be successful, these stochastic processes must match the properties measured empirically in real traffic. The sections below give a mathematical definition of some of these desired properties: long range dependency, self similarity, and heavy-tailed probability distributions.

2.2.2.1. Long-range dependency

A stochastic process is said to be long-range dependent if its autocorrelation function decays hyperbolically. Roughly speaking, in long-range dependent processes correlation between values of the process at different times does not decrease ‘quickly’ as the time difference increases: even when the time difference is high, the correlation between values can be significant.

A discrete-time process, X_t, is long-range dependent if its autocorrelation function satisfies the following property: $R_{XX}(k) \sim k^{2(H-1)}$ for $k \to \infty$, $H \in (0.5, 1]$. The value H is called the Hurst parameter, which is a measure of the correlation of the process. $H = 0.5$ for pure random processes. The closer H is to 1, the higher the correlation of the process, and the longer in range is the dependency.

Long-range dependence is a consequence of self similarity [5], a widely documented effect present in packet traffic (see [5], [9]). There is also criticism of modeling packet traffic using long-range dependence. Clegg, Landa, and Rio [7] state that the impact of the long-range dependence is not relevant to the queuing behavior of the packet traffic. Richard G. Clegg [10] proposes different techniques for measuring the Hurst parameter, H.

2.2.2.2. Self similarity

A process is said to be self similar, or ‘fractal’, when aggregation has no impact on the nature of the process. Roughly speaking, self similarity of a time series means that the series is bursty over several time scales.

Given a discrete-time process,X_t, and the m-aggregated process ${X_t}^{(m)}$ defined as[1]:

$${X_t}^{(m)} = \frac{1}{m} \cdot \sum_{i=tm-m+1}^{tm} X_i \ , \qquad m \in [1,2,\ldots)$$

X_t is said to be self similar if it has the same autocorrelation function as ${X_t}^{(m)}$ for all m.

Self-similarity of Ethernet traffic, and its relation with the notion of ‘burstiness’, was documented by Leland, Taqqu, Willinger, and Wilson [8]. They also state that the aggregation of traffic sources intensifies the self-similarity instead of smoothing it. The reasons behind

[1] ${X_t}^{(m)}$ is a ‘zoom-out’ version of X_t: a higher value of m means a higher time scale.

[2] For example, a voice over IP session where it is known that the CODEC has a 20ms packetization time, can know that there will only be data to transmit every 20 ms and hence the UE need not receive or transmit for

self similarity in World Wide Web (WWW) traffic are studied by Crovella and Bestavros [5]. They explain these reasons in terms of the distributions of WWW document sizes, effects of caching, and user behavior. The last of these causes inactivity periods between downloads of documents.

2.2.2.3. Heavy-tailed distributions

A heavy-tailed probability distribution is one which assigns high probabilities to regions which are far from the mean or the median. Roughly speaking, heavy-tailed distributions are those whose complementary distribution function tends towards zero more slowly than any exponential.

Given a random variable, X, its distribution is heavy-tailed if its complementary distribution function satisfies the following:

$$\bar{F}_X(x) = Pr(X > x) \sim x^{-\alpha} \qquad 0 < \alpha < 2$$

Leland, Taqqu, Willinger, and Wilson point out that superposition of a large number of on/off traffic sources, whose on and off period lengths follow a heavy tailed distribution, generates aggregate traffic which is self similar [9].

2.2.3. Proposed traffic models

The required traffic model for packet switched networks must match the properties of self-similarity and long-range dependence. Some of the proposed models are presented in [4] and briefly described below.

2.2.3.1. On-Off models

In on-off models, a traffic source alternates between two states, ‘on’ and ‘off’. During an on period, traffic is generated at a constant rate, and during an off period there is no traffic. The lengths of on and off periods are independent. If the distribution of these lengths is heavy-tailed, then the superposition of the traffic generated by all the sources will be self-similar and long-range dependent [9].

2.2.3.2. M/G/∞ Processes

It has been shown that as the number of aggregated heavy-tailed on-off sources increases, the resulting process approaches the server occupancy of an M/G/∞ queue in which the service time also follows a heavy tailed distribution [4].

2.2.3.3. Poisson-Pareto Burst Process (PPBP)

The Poisson-Pareto Burst Process (PPBP) uses a model with bursts arriving according to a Poisson process, and whose durations are Pareto distributed. PPBP can be considered as the limiting process for a large number of independent on-off sources aggregated together [4].

3. Methodology

In this chapter we present the abstraction of a ‘packet-burst’, and how we use it to define the different events in the traffic of a user. We also present the different possible predictions to be made about the next events, and their impact.

3.1. Burst model

Smartphones in a mobile network generate packet-based, uplink and downlink traffic. The packet stream generated by each terminal is not continuous over time, i.e., there are some ‘activity’ periods, during which the UE goes to a ‘connected’ state to send and/or receive user information, and there are other periods in which the UE does not send or receive user information at all. Conceptually, a burst can be defined as ‘a period of time during which there is network traffic, separated by periods of time when there is little or no network traffic’. Characterizing user traffic in terms of bursts can help us to understand how activity and inactivity periods are distributed, and consequently, to improve channel scheduling at RNCs. For this purpose, we need a more accurate and precise definition of what a burst is.

The model is structured into different ‘abstraction’ levels. At each level a type of burst is examined. First we consider the packet level, where we will detect 'packet bursts'. At the next abstraction level, we will detect bursts of packet bursts.

3.1.1. Level 0 – Packets

Level 0 is the starting point of the model. We start from a set of packet traces (e.g. from a Wireshark pcap file) captured at the Gi interface. Packet traces consist of timestamped copies of packets ordered by the time the packet was captured.

Each packet trace, P_{UE}, has the following relevant attributes:

‘Owner’ of the trace

UE which generated the trace, i.e., the UE to which the network allocated resources to deliver the packet (if it was a downlink packet) or to allow the UE to send the packet (if it was an uplink packet).

Uplink/Downlink

Uplink packets are those sent from a UE to the network, downlink packets are those the network delivers to a UE.

Relative timestamp t_P

Relative time from when the capture of packets started to when this packet was captured.

Interarrival time IAT_P

Time difference between the relative timestamps of this packet and the immediately previous packet of the same owner and direction (uplink/downlink). If there was no previous uplink/downlink packet associated with this owner, the interarrival time will be taken as zero.

Length l_P	Number of bytes composing the packet.

Figure 4 represents the arrival of packets of a certain owner over the time. The uplink/downlink attribute of the packet is represented as the sign of the length of the packet in the graph: downlink packets are represented with a positive length, while uplink packets are represented with a negative length. Using this representation enables both directions to be presented on a common time axis.

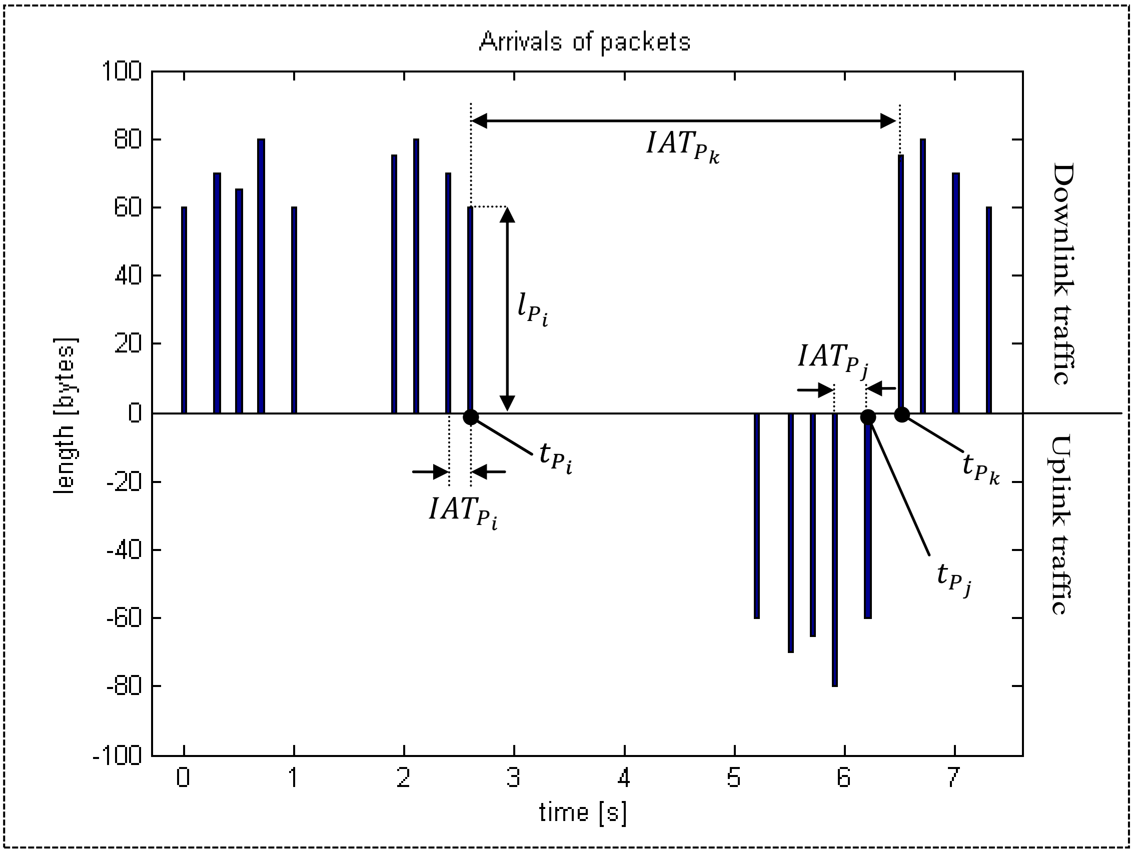


Figure 4: Arrivals of packets of a UE at the Gi interface

In Figure 4 we consider three packets: P_i, P_j, and P_k. The timestamps of these packets are labeled t_{P_i}, t_{P_j}, and t_{P_k}, respectively. Interarrival times of these packets are also labeled, IAT_{P_i}, IAT_{P_j}, and IAT_{P_k}. Note that the interarrival time of a packet is defined in terms of the timestamp of this packet and the immediately previous packet in the same direction. In the case of P_i, P_j, the previous packet in the same direction is the previous packet. In the case of P_k, the previous packet in the same direction is P_i, so $IAT_{P_k} = t_{P_k} - t_{P_i}$. The length of the packet P_i is labeled l_{P_i}.

3.1.2. Level 1 – Uplink/downlink packet bursts

Based on the level 0 information we can define a user's uplink/downlink packet burst as a 'set of consecutive uplink/downlink packets of a certain owner whose interarrival time is less than a certain time threshold'. At this level we maintain the distinction between 'uplink' and 'downlink', because the process for allocation of resources is different when it is triggered by an uplink packet or by a downlink packet.

The relevant attributes of a packet burst, PB, are:

Owner	Owner of the packets composing the packet burst.
Uplink/downlink	Characteristic of the packets composing the packet burst.
Packet burst relative timestamp t_{PB}	Relative timestamp of the first packet composing the packet burst.
Interarrival time IAT_{PB}	Time difference between this packet burst’s relative timestamp and the relative timestamp of the last packet sent or received by this owner before this packet burst.
Packet burst length l_{PB}	Sum of the lengths of all the packets composing the packet burst
Threshold τ_{PB}	Upper bound of the interarrival time of the packets composing the packet burst. Consequently, it is the minimum inactivity time between packet bursts.

If we apply this definition to the packet arrivals shown in Figure 4, taking 0.5 seconds as our threshold value, we can see there are four packet bursts, three of them are downlink packet bursts and one is uplink packet burst. Figure 5 illustrates this.

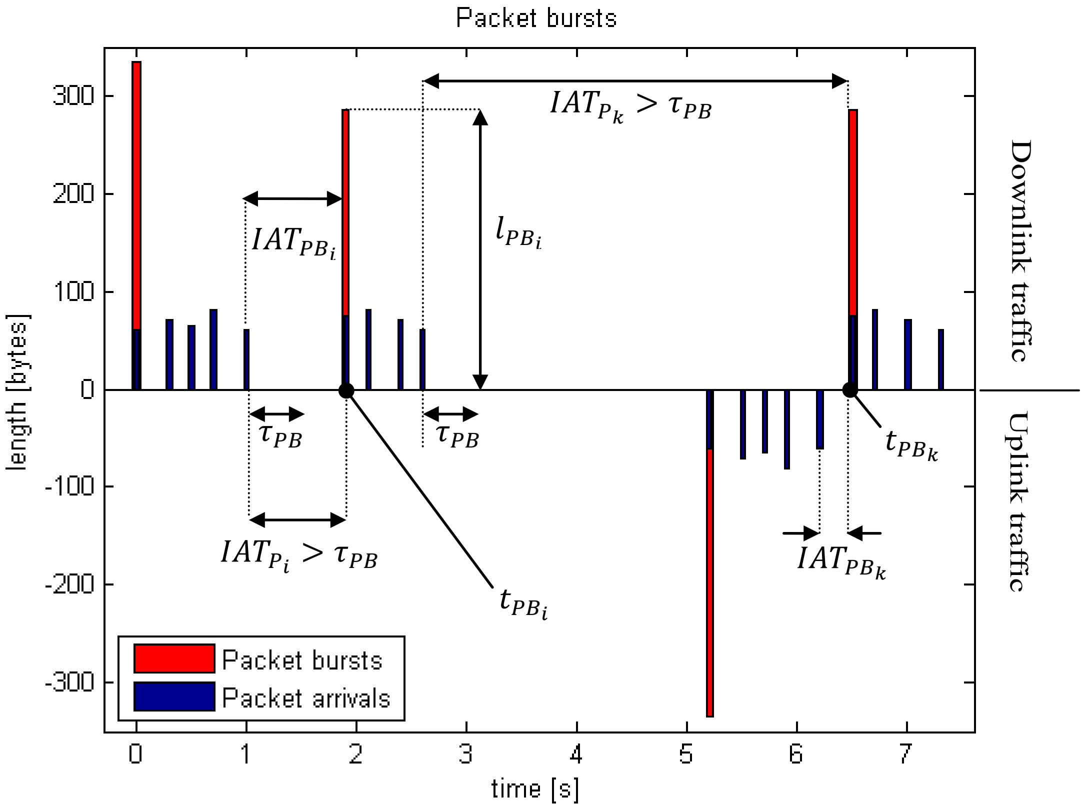


Figure 5: Packet bursts of a UE as seen at the Gi interface

In Figure 5 we note two packet bursts, PB_i and PB_k, with their respective first packets, P_i and P_k. Timestamps of the packet bursts are labeled t_{PB_i} and t_{PB_k}, and coincide with the timestamps of packets P_i and P_k. The packet burst threshold, τ_{PB}, is represented at the end of two packet bursts. Interarrival times of packet bursts are labeled IAT_{PB_i} and IAT_{PB_k}. Note that interarrival time of a packet burst is defined in terms of the last packet before this packet burst, regardless of its direction, while interarrival time of a packet is defined in terms of the last packet in the same direction. This is because the period of inactivity, i.e., the time between the last packet and the arrival of the next packet burst, will be a relevant aspect where we try to improve the use of network resources. If the last packet before a packet burst has the same direction as this packet burst, then the interarrival time of the packet burst and the interarrival time of its first packet will be the same. We can see it in figure 5: $IAT_{PB_i} = IAT_{P_i}$, but $IAT_{PB_k} \neq IAT_{P_k}$. The length of the packet burst PB_i is labeled l_{PB_i}.

3.1.3. Level 2 – Application bursts

Now, based on level 1 information, we can define a user's 'application burst' as a 'set of consecutive packet bursts (either uplink or downlink) of a certain user whose interarrival time is less than a certain threshold'. Application bursts are intended to be a set of packet bursts which are so close in time that is worth to keep network resources allocated to the UE, in order to send/receive them. We will assume that the allocated network resources will be bidirectional, so application bursts will be composed by both uplink and downlink packet bursts.

At this level we have as relevant attributes of each application burst:

Owner
Owner of the packet bursts composing the application burst.

App. burst relative timestamp t_{AB}
Relative timestamp of the first packet burst composing the application burst.

Interarrival time IAT_{AB}
Interarrival time of the first packet burst composing this application burst.

Application Burst length l_{AB}
Sum of the packet burst lengths of all the packet bursts composing the application burst.

Threshold τ_{AB}
Upper bound of the interarrival time of the packet bursts composing the application burst. Consequently, it is the minimum interarrival time of application bursts.

Applying this definition to the packet bursts shown in Figure 5, taking 2 seconds as the threshold value, we can see there are two application bursts. The length of the application bursts is considered always positive, since in this level there is no distinction between uplink and downlink bursts.

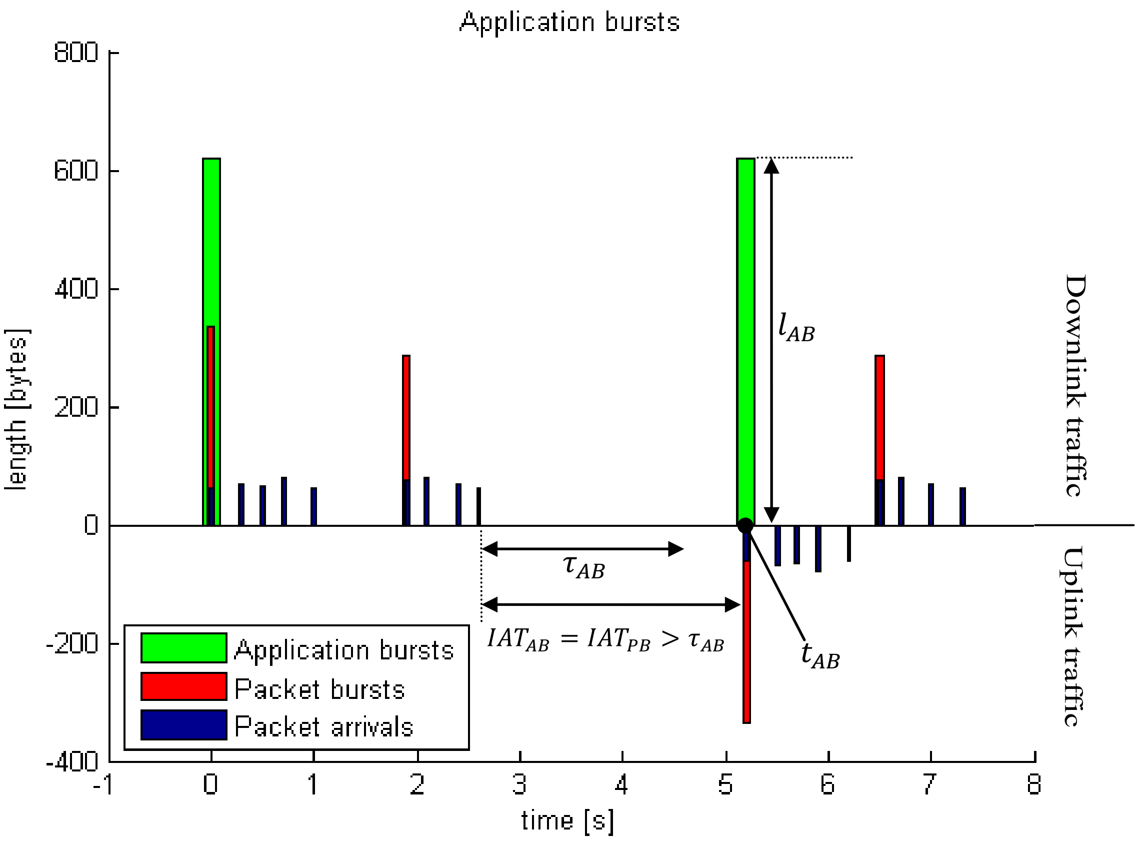


Figure 6: Application bursts of a UE

3.1.4. Level n – Generalization

The model can be generalized to even higher levels: a 'level n burst' is a set of 'level n-1 bursts' whose interarrival time is less than a certain threshold. For example, 'session bursts' can be defined, following this pattern, as bursts of application bursts.

At each level, the 'time threshold' value is a parameter of the model, and it defines the 'minimum inactivity period' between level n-1 bursts used to identify level n bursts. Time thresholds of level n are always greater than of level n-1.

Level n bursts have information about their owner, relative timestamp, and length based on information of lower levels. Interarrival time of level n bursts is defined within level n.

3.2. Events at packet burst level

We can learn more about the traffic of a user if we view it as a sequence of events related with the ‘packet burst’ level of abstraction defined previously in section 3.1.2 (level 1).

3.2.1. Naive approach

The packet traffic of a user, at the packet burst level, is a sequence of events. An event is one of the following:

1. Arrival of an uplink packet burst,
2. Arrival of a downlink packet burst, or
3. Inactivity: no arrival of uplink or downlink packet burst for a period of time of, at least, the application burst threshold (τ_{AB}). An inactivity event implies the end of the current application burst.

Figure 7 illustrates this. It represents the same packet bursts as shown in Figure 5. We can see that the sequence of events for this user is: downlink packet burst, downlink packet burst, inactivity, uplink packet burst, downlink packet burst.

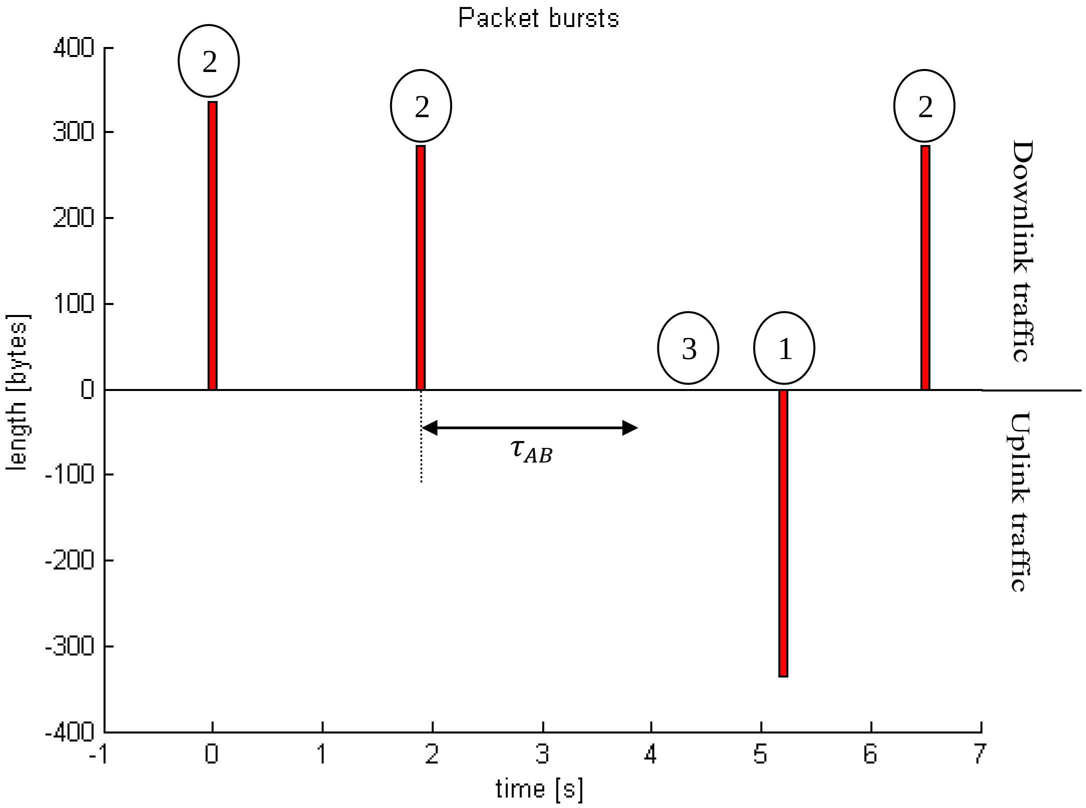


Figure 7: Events at packet burst level, naive approach

The events 'uplink packet burst' and 'downlink packet burst' are considered 'activity events'. They require the UE to be in Cell_FACH or Cell_DCH state to transmit or receive packets of user data. As per section 2.1, when the amount of data to be transmitted to or from the UE is significant, a DCH is allocated for the user. A DCH provides higher bit rates, but also has higher battery power consumption by the terminal. Another drawback is that the number of DCHs a RNC can allocate is limited.

We will assume that the packet bursts require the allocation of a DCH, and it is kept allocated until the transmission of the last packet in this burst. When a packet burst arrives and the UE is not in Cell_DCH state, there will be increased latency due to the time required for the state change and the allocation of the DCH. The RNC associates 'inactivity timers' with each DCH, where 'inactivity' means no transmission of packets in either the uplink or the downlink direction. After the transmission of the last packet of a packet burst, the channel is not released until the associated inactivity timer expires, so bursts arriving within that time do not experience extra latency. The application burst threshold of our burst model, τ_{AB}, is intended to represent the duration of inactivity caused by these inactivity timers.

This is a naive approach because it does not take into account that, in reality, packet bursts are not instantaneous. They have duration in time, i.e., the difference between the timestamps of the last and the first packet in the packet burst. This means that an uplink packet burst can overlap a downlink packet burst.

3.2.2. A more realistic approach

In a more realistic approach, the inactivity event needs to be redefined as follows: no arrival of an uplink or downlink packet burst within the time from the arrival of the last packet, for a period of time of, at least, the application burst threshold (τ_{AB}). This is a more realistic approach, because it considers the duration in time of packet bursts in order to capture more exactly when the inactivity of the channel starts.

Figure 8 illustrates this more realistic approach to events at the packet burst level, which requires a redefinition of the inactivity event. The sequence of events in this case is: downlink packet burst, uplink packet burst, inactivity, uplink packet burst, downlink packet burst. The first downlink packet burst overlaps with the first uplink packet burst; the order is determined by the timestamp of the first packet of the burst.

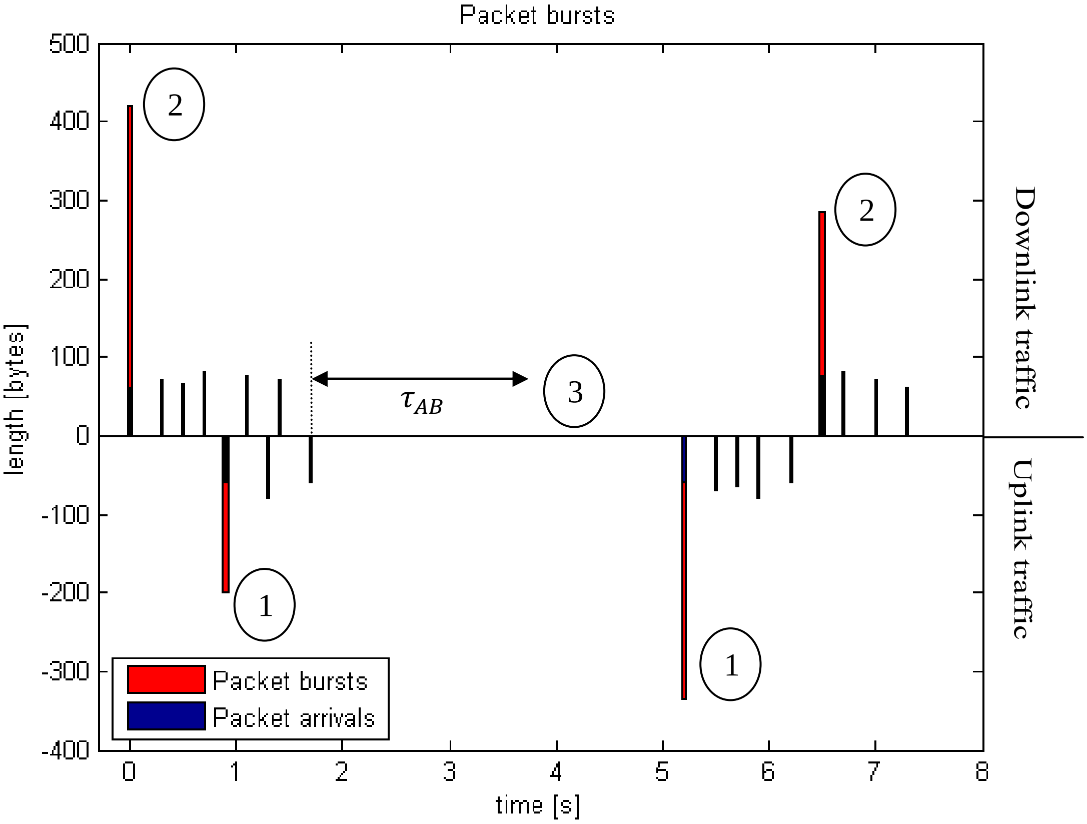


Figure 8: Events at packet burst level, a more realistic approach

3.3. Predictions about packet burst events

Instead of always relying on timers to release resources, it would be more efficient for the network to explicitly decide whether it should keep or release the channel allocated to the UE, based on a prediction of which of the three possible events in the user traffic will occur next.

3.3.1. Prediction moments

The network has to decide between releasing or keeping resources dedicated to a user when the end of a packet burst is detected and there is no transmission of a packet burst in the opposite direction. We call this time a ‘prediction moment’. The end of a packet burst will be detected after τ_{PB} (packet burst threshold) seconds without a packet transmission in the direction of the burst. Therefore, a prediction moment will occur after τ_{PB} seconds without any packet arrival, via either the uplink or downlink.

Figure 9 shows three prediction moments in the traffic of a user. At every prediction moment it is clear what the previous events are. For the two first prediction moments, $t_{pred,1}$ and $t_{pred,2}$, it is also clear what is the next event. This is not clear for the third prediction, at $t_{pred,3}$, as the next event could be inactivity or a packet burst (which actually occurs, but we do not know because these packets have not yet been captured in the packet trace).

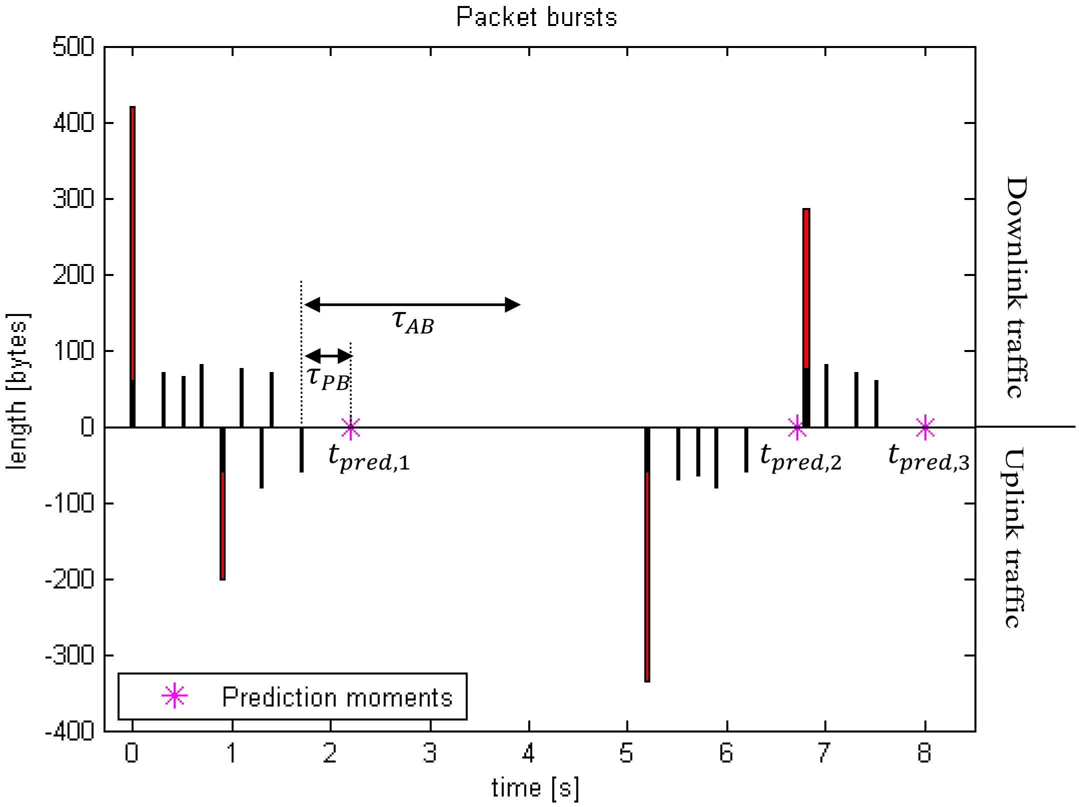


Figure 9: Prediction moments

3.3.2. Consequences of predictions

At every prediction moment, the network will predict the next event in the traffic. Such a prediction will have the following consequences.

Prediction of activity event (uplink or downlink packet burst): If the network thinks that the next packet burst will come within τ_{AB} seconds, it should keep the resources allocated for this user. If the prediction is correct, then the next burst will not suffer latency due to the state switching, as the resources for transmission are already allocated. On the other hand, if the prediction is incorrect, there will be τ_{AB} seconds of inactivity while the necessary resources are still allocated, but they will be released after that time. Therefore, if the prediction is correct there will be a small waste of resources - since the resources are unused for a small period of time, but the traffic will not be delayed. Whereas if the prediction is incorrect, the resources can be allocated to another UE, leading to increased signaling overhead and increased delay for the traffic to the UE.

Prediction of inactivity event: If the network thinks that there will be an inactivity time of at least τ_{AB} seconds, then the network should release the resources allocated for this UE. If the prediction is correct, then the network will have avoided wasting these resources for the inactivity time, and the UE will not waste battery power as it can power off its receiver. However, if the prediction is incorrect, then the next burst will suffer increased latency due to the need to reallocate resources for this UE. In addition, if the prediction is wrong there will be unnecessary signaling overhead.

As a result of these predictions, the UEs which can take the greatest advantage of these predictions are those whose packet traffic alternates activity events with inactivity events. For these users, a correct prediction of the inactivity events will remove most of the inactivity time from their allocation of dedicated resources. This saving due to correct predictions is especially true for UEs for which the activity or inactivity times are long and periodic - as the predictions can potentially be correct a large fraction of the time (assume that suitable levels of this traffic's characteristics are considered)[2]. Users who continuously send and/or receive packet bursts will not benefit from predictions, as they will always keep the resources as the resources are in use. Other considerations may apply to these users who are always transmitting or receiving, such as the unfairness of a single user being assigned resources for a long period of time. However, these other considerations are out of the scope of these predictions.

3.3.3. Probabilities for predictions

Predictions are made based on certain probability values:

p_A : probability that the next event is an activity event (either downlink or uplink packet burst).

p_{IA} : probability that the next event is an inactivity event. $p_{IA} = 1 - p_A$

The values for these probabilities should be such that the number of correct predictions is as high as possible, in order to achieve the maximum benefit. Events in the traffic are usually correlated, due to the higher layer protocols, so it is logical to think that the values of p_A and p_{IA} depend on patterns in the recent events for this traffic. Consequently, these values are not static in time; but rather they should be recalculated at every prediction moment based on past events. Note that if all predictions are made taking $p_A = 1$, then we will have a system as the described in section 2.1.4, in which transitions to lower power consumption states and release of resources are controlled exclusively by inactivity timers. On the other hand, if all predictions are made taking $p_A = 0$, resources will always be released as soon as inactivity is detected, achieving the highest possible reduction in resources use during the inactivity time; but this will also maximize the number of packet bursts suffering increased latency.

[2] For example, a voice over IP session where it is known that the CODEC has a 20ms packetization time, can know that there will only be data to transmit every 20 ms and hence the UE need not receive or transmit for this period of time. See [11].

At this point, we can reconsider one of our first high level research questions (presented in subsection 1.1.3), ‘What information can be extracted from the presence of patterns in the traffic?’. We can reformulate this question as: ‘Can we extract from the presence of patterns information in order to calculate values for p_A and p_{IA} that maximize the number of correct predictions?’.

3.3.4. Estimation of p_A

At every prediction moment, a value of p_A is estimated based on two elements: an event buffer and a table.

The event buffer is an ordered list of the last k events of the UE, between the last inactivity event and the current activity event. The buffer will have capacity to store n events. Here, k and n are positive integers, and $n \geq k \geq 1$.

The table has one row per each possible sequence and subsequence of events stored in the buffer, and three columns. There is a one-to-one correspondence between rows in the table and sequences of events. To simplify notation, ‘row k’ is the row corresponding to the sequence of the k events stored in the buffer; ‘row $k-1$’ is the row corresponding to the sequence of the last $k-1$ events, etc. Each column stores the number of times the sequence of events corresponding to that row was followed by an uplink burst, a downlink burst, or inactivity event. We call these numbers $m_{UL,k}$, $m_{DL,k}$ and $m_{I,k}$, respectively.

Every time a new event is detected, the event buffer and the table are updated. In the table, at rows k, $k-1$, ..., 2, and 1, the value stored in the column corresponding to the detected event is incremented by one unit. After that, the event buffer is updated: the detected event is added and, if the capacity of the buffer is reached, the oldest event is discarded. If the detected event is inactivity, then all the stored events are discarded and the inactivity event is added.

The value of p_A at every prediction moment is calculated as:

$$p_A = \begin{cases} \dfrac{\sum_{i=1}^{k} m_{UL,i} + m_{DL,i}}{\sum_{i=1}^{k} m_{UL,i} + m_{DL,i} + m_{I,i}}, & \displaystyle\sum_{i=1}^{k} m_{UL,i} + m_{DL,i} + m_{I,i} \neq 0 \\ 1, & \displaystyle\sum_{i=1}^{k} m_{UL,i} + m_{DL,i} + m_{I,i} = 0 \end{cases}$$

Where k is the number of events stored in the buffer at the prediction moment, and values $m_{UL,i}, m_{DL,i}$, and $m_{I,i}$ are extracted from row i of the table. If the sum of these values is 0, then there is no information to predict the next event based on previous events, so a default value of 1 for p_A is taken, and the inactivity timer will control the release of resources if there is no activity.

3.3.5. Value of τ_{AB}

Two alternatives are considered in order to set the value of the parameter τ_{AB}. The first is to set τ_{AB} to a static value, and the second is to dynamically calculate a suitable value for τ_{AB}, based on particularities of each user's traffic. These alternatives are detailed below.

The 'static' alternative means that every user at every moment will have τ_{AB} set to the same value. This has some drawbacks: if at a prediction moment the system predicts activity for a certain user, and packet bursts associated with this user arrive shortly after τ_{AB}, then these bursts will suffer a delay due to the overhead of reallocating resources, which could easily have been avoided by keeping the resources allocated for this user for a few more milliseconds. If the interarrival time of packet bursts of this user is usually below τ_{AB}, it may be better to set this parameter to a lower value for this user.

Using the dynamic alternative, a new value for τ_{AB} will be calculated at every prediction moment. This calculation will be made based on previous values of interarrival times of packet bursts of this user. This makes τ_{AB} user-dependent and time-dependent. Every time a new packet burst is detected, its interarrival time is checked, and if it is below a maximum value, $IAT_{PB,max}$, it will be stored. If the interarrival time of the new detected packet burst is higher than $IAT_{PB,max}$, we will discard all previously stored values, if any, since $IAT_{PB,max}$ will be high enough to expect that the circumstances of the user may have changed (e. g., the user has moved to a place with different connectivity quality, that will affect the bit rate of the air interface and, consequently, the interarrival time of packet bursts seen at the Gi interface). When a prediction moment comes, a value of τ_{AB} is calculated as the average value of the last h stored values of interarrival times of packet bursts (h is a positive integer). If there are no values stored, then τ_{AB} will be set to a default value, $\tau_{AB,def}$.

3.3.6. Evaluation

The gains we get from this prediction system are measured in terms of reduction of the amount of time when the UE is inactive while in Cell_DCH state, and cost paid in terms of reduction of packet burst throughput. These metrics are computed relative to the inactivity time and the burst throughput of a system as described in section 2.1.4, where transitions to lower power consumption states of the UE and release of resources are controlled exclusively by inactivity timers or, equivalently, a system where $p_A = 1$ at every prediction moment.

Every prediction moment, $t_{pred,i}$, implies occurrence of an inactivity time, $t_{IA,i}$. The minimum value for $t_{IA,i}$ is τ_{PB} seconds, due to how the prediction moment is defined (see figure 9). The rest of the inactivity time depends on the decision taken at the prediction moment about keeping the UE in Cell_DCH state or transitioning to lower consumption states. Due to this, the probability of activity is calculated for the prediction moment, $p_{A,i}$. This time will also depend on the expected interarrival time of the next packet burst $IAT_{PB,i+1}$ (i.e., the time between the arrival of the last detected packet, $t_{P,i}$, and the arrival of the next packet burst, $t_{PB,i+1}$) and on whether it is higher than the value of the application burst threshold at the prediction moment, $\tau_{AB,i}$, or not.

$$\begin{aligned} t_{IA,i} &= \tau_{PB} + \left(\min\left(\tau_{AB,i} - \tau_{PB}\ , t_{PB,i+1} - t_{P,i} - \tau_{PB}\right)\right) \cdot p_{A,i} \\ &= \tau_{PB} + \left(\min\left(\tau_{AB,i} - \tau_{PB}\ , IAT_{PB,i+1} - \tau_{PB}\right)\right) \cdot p_{A,i} \end{aligned}$$

The throughput of a packet burst PB, R_{PB}, is a function of its length (l_{PB}), the time of transmission/reception of the burst ($t_{TX/RX}$), the time it takes to change the UE to Cell_DCH state, t_{switch}, and the probability that the UE is not in Cell_DCH state at the moment the burst arrives (p_{DCH_off}).

$$R_{PB} = \frac{l_{PB}}{t_{switch} \cdot p_{DCH\_off} + t_{TX/RX}}$$

The time of transmission/reception depends on the length of the burst and the bit rate (b):

$$t_{TX/RX} = \frac{l_{PB}}{b}$$

The value of p_{DCH_off} depends on the interarrival time of the packet burst, $IAT_{PB} = t_{PB} - t_P$, where t_{PB} is the timestamp of the packet burst, and t_P is the timestamp of the last packet before the packet burst. If this difference is greater than τ_{AB}, it is certain that the UE will not be in Cell_DCH, as the inactivity timer will have triggered a transition to a state of lower power consumption. On the other hand, if the difference is less than τ_{PB}, it means that there was a transmission or reception of packet burst in the other direction, so the UE will already be in Cell_DCH state. If the difference is between τ_{PB} and $\tau_{AB,i}$, this means that the burst arrived just after a prediction moment, and the status of the UE depends on the decision taken, therefore, it depends on the value of $p_{A,i}$ calculated for that prediction moment.

$$p_{DCH\_off} = \begin{cases} 1, & IAT_{PB} > \tau_{AB} \\ 1 - p_{A,i}, & \tau_{PB} < IAT_{PB} \leq \tau_{AB} \\ 0, & IAT_{PB} \leq \tau_{PB} \end{cases}$$

Given the packet traces of a UE, the total inactivity time is obtained as the sum of inactivity times at every prediction moment, excluding the last prediction moment, because it is impossible to know when the next packet burst would arrive. Also, a mean value of the burst throughput is calculated considering as the throughputs of all the packet bursts detected.

The goal of this prediction system is to reduce the inactivity time when the UE is in Cell_DCH state, without reducing the burst throughput. Success will be strongly related to the number of correct predictions. For every prediction moment, $t_{pred,i}$, there will be a rate of correct predictions, which will be $p_{A,i}$, if the next event after the prediction moment is an uplink or downlink packet burst, or $p_{IA,i} = 1 - p_{A,i}$, if the next event was inactivity. The mean value of the rates of correct predictions of all the prediction moments of the UE will be calculated to explore the relationship between correct predictions and gains obtained from the system.

3.3.7. Processing of packet traces

Given a set of packet traces of a certain UE, it is necessary to extract the required data to make predictions and to perform an the evaluation of the prediction system. These data are:

- The set of events at the packet burst level. This is necessary both to make the predictions, since the estimation of the values of p_A is based on it, and to evaluate the system, because it will allow us to know the next event after a prediction moment and compute the probability of correct predictions.
- The set of packet bursts transmitted and received by the UE: timestamp, direction (uplink or downlink), and length. This is necessary for the calculation of burst throughputs, on which evaluation of the system is based.
- For every burst, the value of p_{DCH_off} is the probability that the burst arrives when the UE is not in Cell_DCH state. This value is computed in order to obtain the burst throughputs.
- The prediction moments detected: when to predict ($t_{pred,j}$) and the value of $p_{A,j}$ is calculated for each prediction.
- If the parameter τ_{AB} is set dynamically at every prediction moment, then we want to know the value calculated for it at every prediction moment, $\tau_{AB,j}$
- Value of the inactivity time of the UE in Cell_DCH state. This will be used to evaluate the system.

In order to get this information, every packet trace of the UE, P_i, is processed. We will assume that packets in each trace are in ascending order of their timestamps. To process packet i (P_i) we extract its timestamp, t_{P_i} , its length, l_{P_i} , and its direction, uplink or downlink. We also keep track of the previously processed packets, specifically:

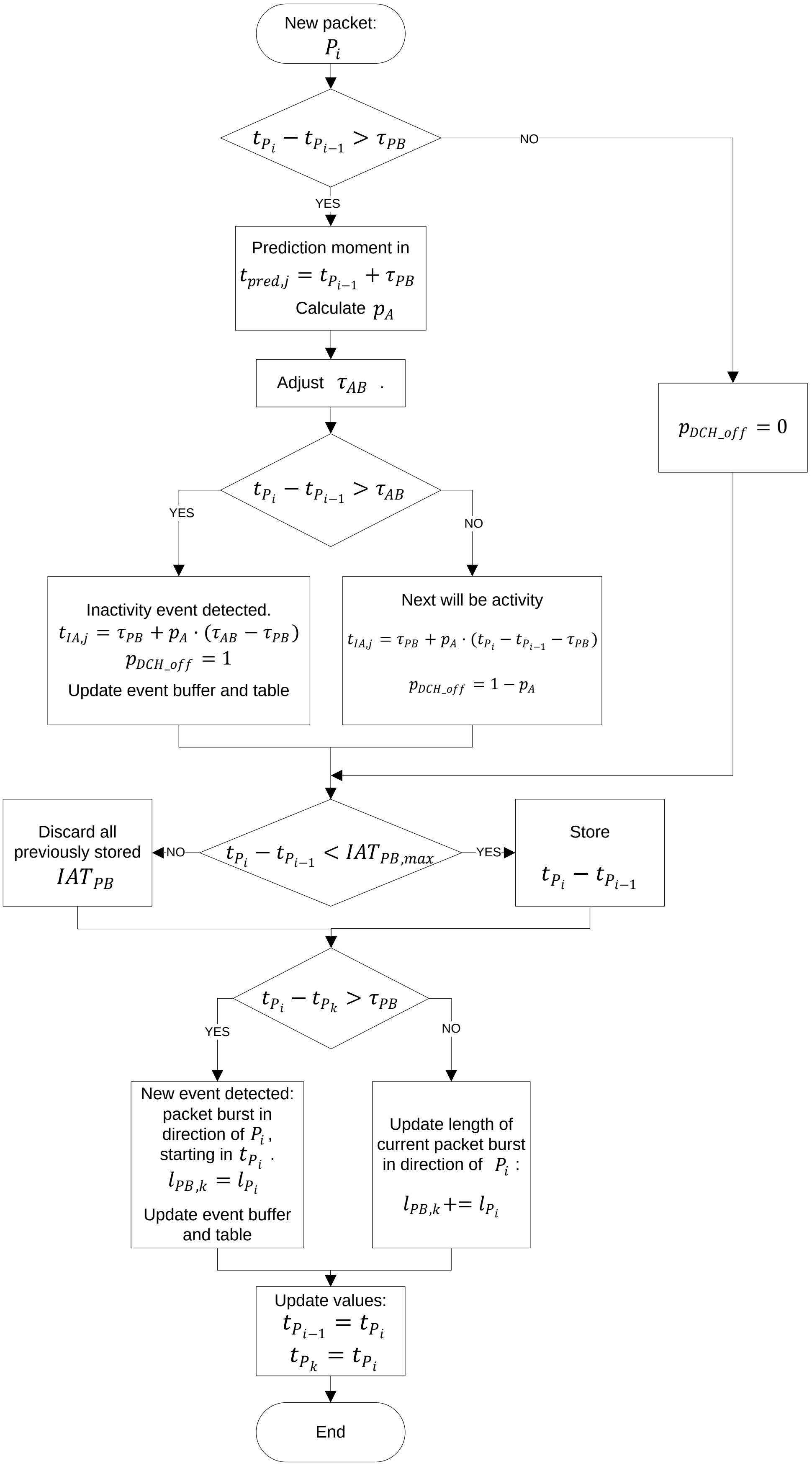
$t_{P_{i-1}}$ Timestamp of the immediately previous processed packet.

t_{P_k} Timestamp of the last processed packet whose direction was the direction of P_i. Since direction of P_i can be uplink or downlink, we need to keep the timestamps of the last uplink packet processed, $t_{P,UL}$, and the last downlink packet processed, $t_{P,DL}$.

$l_{PB,k}$ Cumulative sum of lengths of packets composing the last detected packet burst in the direction of P_i. This yields two values: length of the last detected uplink packet burst, $l_{PB,UL}$, and length of the last detected downlink packet burst, $l_{PB,DL}$.

Finally, the processing of a packet trace requires an event buffer of size n, values for the packet burst threshold, τ_{PB}, and the application burst threshold, τ_{AB} (if static). If τ_{AB} is calculated dynamically, then the system needs values for the maximum packet burst interarrival time, $IAT_{PB,max}$, for the default value of the application burst threshold, $\tau_{AB,def}$, and for the number of previous interarrival times of packet bursts to be considered in the calculation, h (see section 3.3.5).

The processing of a packet trace is shown in the flowchart in figure 10. The process represented is generic, allowing both ‘static’ and ‘dynamic’ ways to set the parameter τ_{AB}. In the first case, the process ‘Adjust τ_{AB}’ will set it always to the same value, while in the second it will take into account the stored values of interarrival times of packet bursts to set the parameter to the average value of the last h stored values or to the default value. We will process all the sets of packet traces of the different UEs. After that, we will have all the necessary to evaluate the prediction system.



New packet: P_i
$t_{P_i} - t_{P_{i-1}} > \tau_{PB}$
NO
YES
Prediction moment in $t_{pred,j} = t_{P_{i-1}} + \tau_{PB}$
Calculate p_A
Adjust τ_{AB} .
$p_{DCH\_off} = 0$
$t_{P_i} - t_{P_{i-1}} > \tau_{AB}$
YES
NO
Inactivity event detected.
$t_{IA,j} = \tau_{PB} + p_A \cdot (\tau_{AB} - \tau_{PB})$
$p_{DCH\_off} = 1$
Update event buffer and table
Next will be activity
$t_{IA,j} = \tau_{PB} + p_A \cdot (t_{P_i} - t_{P_{i-1}} - \tau_{PB})$
$p_{DCH\_off} = 1 - p_A$
Discard all previously stored IAT_{PB}
NO
$t_{P_i} - t_{P_{i-1}} < IAT_{PB,max}$
YES
Store $t_{P_i} - t_{P_{i-1}}$
$t_{P_i} - t_{P_k} > \tau_{PB}$
YES
NO
New event detected: packet burst in direction of P_i, starting in t_{P_i} .
$l_{PB,k} = l_{P_i}$
Update event buffer and table
Update length of current packet burst in direction of P_i :
$l_{PB,k} {+}{=} l_{P_i}$
Update values:
$t_{P_{i-1}} = t_{P_i}$
$t_{P_k} = t_{P_i}$
End

Figure 10: Processing of a packet trace

3.3.8. Further predictions

As it was previously said, an inactivity event implies the end of an application burst. When an inactivity event happens, additional predictions can be made about the next application burst.

Predictions of the inactivity duration. The interarrival time of the next application burst could be predicted, to estimate how long the UE will be inactive, in order to determine when it is probable that the UE will transmit or receive data. Assuming the self similarity of the aggregated traffic, the UE could be modeled as an on/off source, with a heavy-tailed distribution of on and off periods (similar to the models mentioned in [4] and [9]). Parameters for the distribution of the 'off' period (inactivity) durations could be calculated based on previous durations of off periods of this UE.

Predictions of the length of the next application burst. These predictions would help to decide whether to dedicate a channel for this UE. If the predicted length is above a certain threshold, then a dedicated channel shoud be allocated, otherwise the UE should transmit or receive through shared channels. Using the same assumptions and models as for predictions of inactivity duration, and accepting the explanation of the causes of the heavy-tailed distribution of 'on' period durations given in [5], we could model the length of the application bursts with a Pareto distribution. Parameters for the distribution of application burst lengths could be calculated from previous application burst lengths of this UE.

3.4. Predicting interarrival time

In this section we describe a method to estimate an upper bound of the interarrival time of every packet sent or received by a UE. The definition of interarrival time of a packet here is the time between the arrival of the immediately previous packet associated with the same UE and this packet, regardless of the direction of both packets.

For every UE, we will have a sequence, $IAT[n]$, defined as the interarrival time of the n-th packet. Interarrival time in this section is defined independent of the direction of the packets. The sequence $IAT[n]$ will capture interarrival times of uplink and downlink traffic, and will enable us to compute possible correlations between them.

This methodology is described for packets, but it can be generalized to any of the abstractions introduced in section 3.1. Knowing an upper bound of the time that is going to pass between the arrival of a packet (or a packet burst) allows us to set the values of the thresholds (τ_{PB}, τ_{AB}) to appropriate values.

3.4.1. Distributions of interarrival time

We consider the sequence of interarrival times of packets to/from a certain UE. $IAT[n]$ is the time between the $(n-1)$-th and the n-th packet observed of this UE, regardless the direction of the packets. $IAT[n]$ will be a positive real number ($IAT[n] \in \mathbb{R}^+$). In the traffic a UE generates in a real UMTS network it is observed that the values of $IAT[n]$ follow the distribution shown in Figure 11 (see section 4.1 for detailed information about the traffic analyzed).

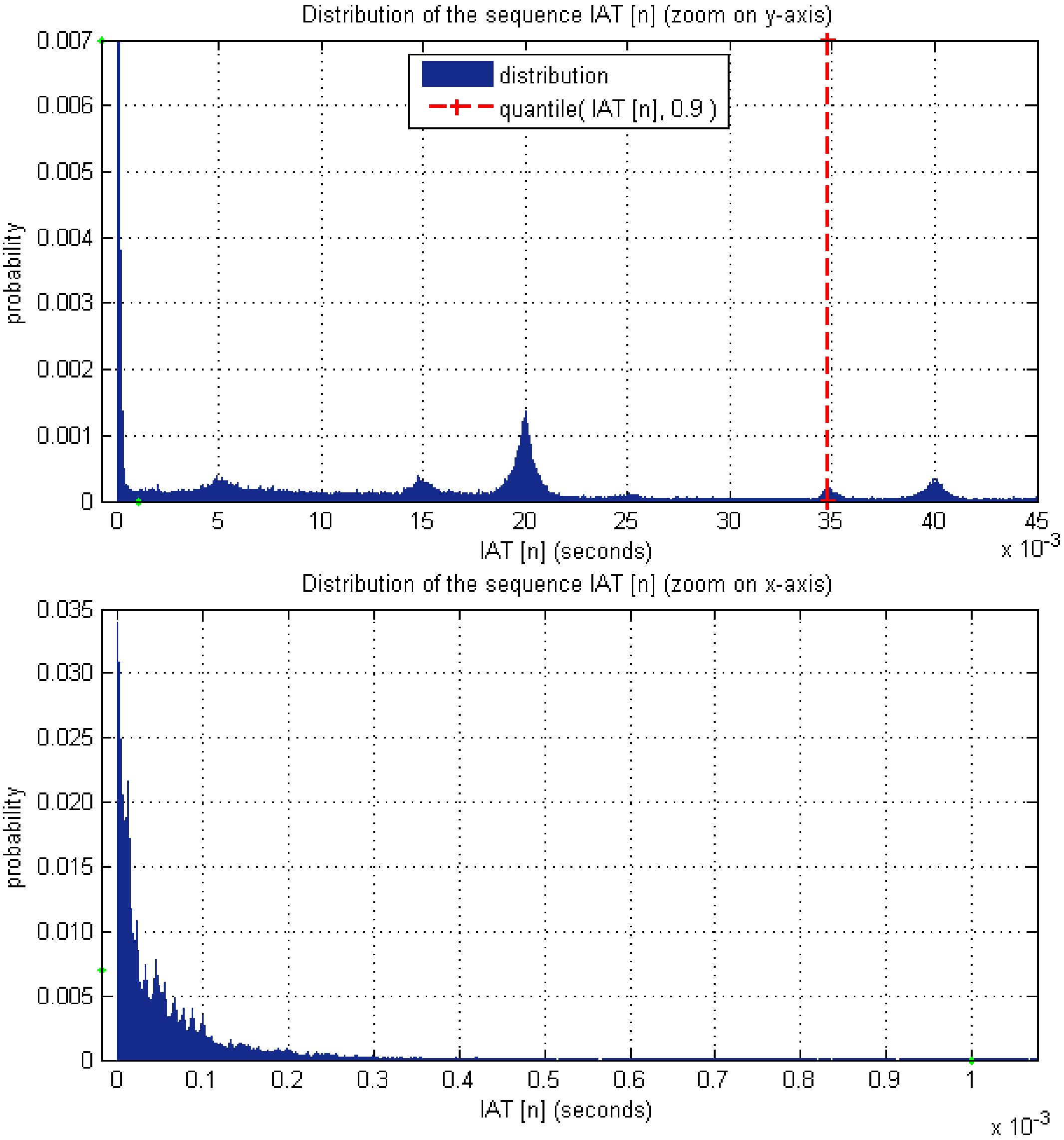


Figure 11. Distribution of interarrival times of packets: (a) zoom on y-axis, (b) zoom on x-axis.

The red vertical line in the upper portion of Figure 11 represents the value of the quantile of $IAT[n]$ for a probability of 0.9. This value is 0.0348 seconds, and it means that 90% of the packets arrive less than 0.0348 seconds after the arrival of the previous packet. If we select this value as upper bound of the interarrival time of the next packet, we will fail to include the 10% of the packets, according to the distribution. We could cover a higher percentage of the interarrival times selecting a higher value for the upper bound, but we would be overestimating for most of the interarrival times.

The question now is it is possible to obtain more accurate upper bounds for the interarrival time of the next packet, keeping this probability of success, based on previous values of the sequence $IAT[n]$. Figure 12 shows distributions of the interarrival time conditioned to the value of the interarrival time of the immediately previous packet from this UE, considering four different ranges for $IAT[n-1]$.

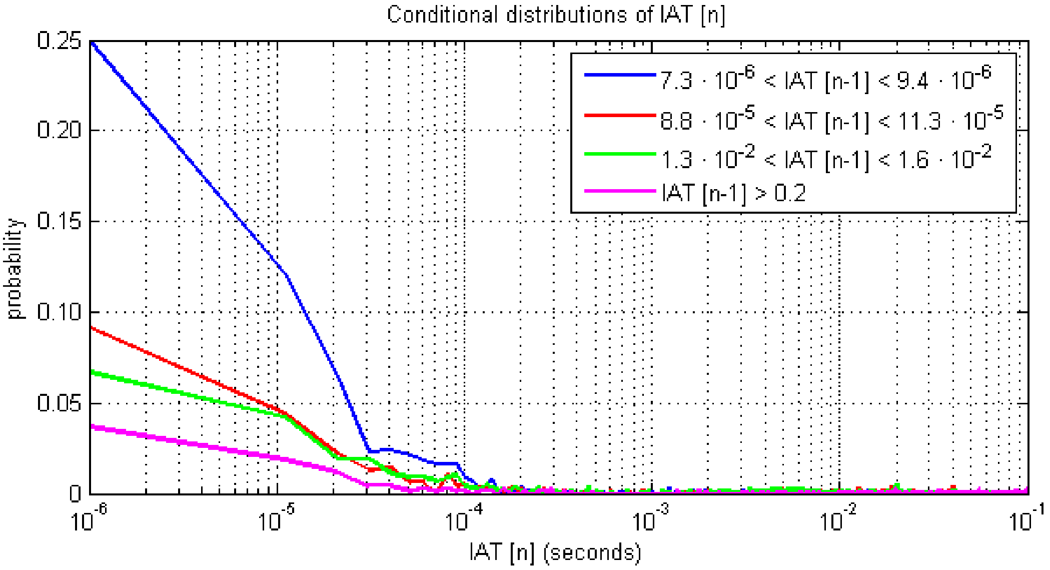


Figure 12: Distributions of $IAT[n]$ conditioned to $IAT[n-1]$

Figure 12 illustrates that the distributions of $IAT[n]$ are different when the values of $IAT[n-1]$ belong to different ranges. Hence, quantiles of $IAT[n]$ conditioned to $IAT[n-1]$ are different, depending on which range $IAT[n-1]$ belongs to. Table 1 shows quantiles for a probability of 0.9 of different conditional distributions of $IAT[n]$.

Table 1: Quantiles for probability 0.9 of the conditional distributions of $IAT[n]$

Range of $IAT[n-1]$			**quantile ($IAT[n]$, 0.9)**
lower bound	**upper bound**	**color**	
$7.3 \cdot 10^{-6}$	$9.4 \cdot 10^{-6}$	Blue	0.0203
$8.8 \cdot 10^{-5}$	$11.3 \cdot 10^{-5}$	Red	0.0398
$1.3 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	Green	0.0383
0.2	∞	Magenta	1.9078

3.4.2. Upper bounds for interarrival time

Every time a packet arrives, an upper bound is estimated for the interarrival time of the next packet. This estimated upper bound will be the quantile for a certain probability, p, of the distribution of $IAT[n]$, conditioned to the values of the previous k samples of the sequence. The estimation requires a partition of the set of positive real numbers, $\mathbb{R}^+$, and a set of vectors containing samples of $IAT[n]$. This process is described below.

Let $\{T_i\}, i = 1,2, \dots, j$ be a partition of $\mathbb{R}^+$ into j intervals. Since it is a partition, for all n there will be a value of i such that $IAT[n] \in T_i$. Before the arrival of the n-th packet, a buffer of size k stores the values of i corresponding to $IAT[n-k], \dots, IAT[n-1]$. When the n-th packet arrives, the buffer is updated with the value of i corresponding to $IAT[n]$.

There will be a set V of j^k vectors, and a one-to-one correspondence between every possible status of the buffer and every vector in V. When the n-th packet arrives, the value $IAT[n]$ is added to the vector corresponding to the status of the buffer before the arrival. Hence, every vector has samples of the random variable $IAT[n]$ conditioned to the values of

$IAT[n-k], \dots, IAT[n-1]$. The maximum number of samples stored in every vector is limited to m.

After the arrival of the n-th packet, once the buffer is updated, an upper bound is estimated for the value of the interarrival time of the next packet, $IAT[n+1]$. This is made by accessing the vector corresponding to the status of the updated buffer and calculating the value of the quantile of the samples stored in it for the given probability p.

Summarizing, the parameters involved to estimate an upper bound for $IAT[n+1]$ are:

p Probability used to calculate quantiles. Ideally, p will be the portion of packets whose interarrival time is below their estimated upper bound.

k Number of previous packets considered to estimate the conditional distribution of $IAT[n]$. It determines the size of the buffer and the number of vectors in V.

$\{T_i\}$ Partition of $\mathbb{R}^+$ used to classify values of $IAT[n]$. Its cardinality, j, along with k, determines the cardinality of V.

V Set of vectors containing samples of $IAT[n]$ conditioned to previous values of the sequence. There will be j^k vectors.

m Maximum number of samples stored in every vector in V.

3.4.3. Evaluation

It is desirable that the number of packets whose interarrival time is below its estimated upper bound is as high as possible, while the estimated upper bounds are as low as possible. If the second condition was not imposed, the first could be easily achieved by overestimating the upper bound, but this would not give us accurate bounds. The aim of using the distribution of $IAT[n]$ conditioned to the previous k values is to improve the accuracy of the estimated upper bounds.

For every set of packets to/from a UE, we will evaluate the percentage of packets whose interarrival time is below its respective estimated upper bound. This percentage ideally will be p. We will also evaluate the average value of all the estimated upper bounds.

Evaluations will be made for different values of k, starting at 0, which would be equivalent to use the distribution of $IAT[n]$ not conditioned to any previous value. Increasing k, the number of vectors in V increases. Consequently, for a given number of packets, increasing k means that the number of samples stored in each vector in V decreases (while the maximum capacity of the vector has not been reached). Hence, increasing k we have less precision when calculating quantiles. To capture the benefit of using a value of $k > 0$, for each packet of the UE, we will estimate two upper bounds for $IAT[n+1]$. The first bound will be estimated exactly as described in section 3.4.2: getting all the samples of $IAT[n]$ conditioned to the previous k values, available in a vector of V, and calculating their quantile for the probability p. For the second, we will get the same number of samples of $IAT[n]$, but not conditioned to previous values: we will consider $IAT[n], IAT[n-1], IAT[n-2]$ ..., until we get the same number of samples as the stored in the considered vector of V, and then we will calculate their quantile for the probability p. Since both upper bounds are estimated using the same number of samples, they have the same precision, and the difference between them will only be due to the fact of using the conditional distribution of $IAT[n]$ (a value of $k > 0$). Then, for both sets of upper bounds, we will evaluate the percentage of interarrival times below their respective upper bound (this should be the same for both sets of upper bounds, ideally p) and the average value of the estimated upper bounds (this should be lower for the set of bounds got from conditional distributions).

We also evaluate the effect of using different partitions of $\mathbb{R}^+$: linear and logarithmic partitions.

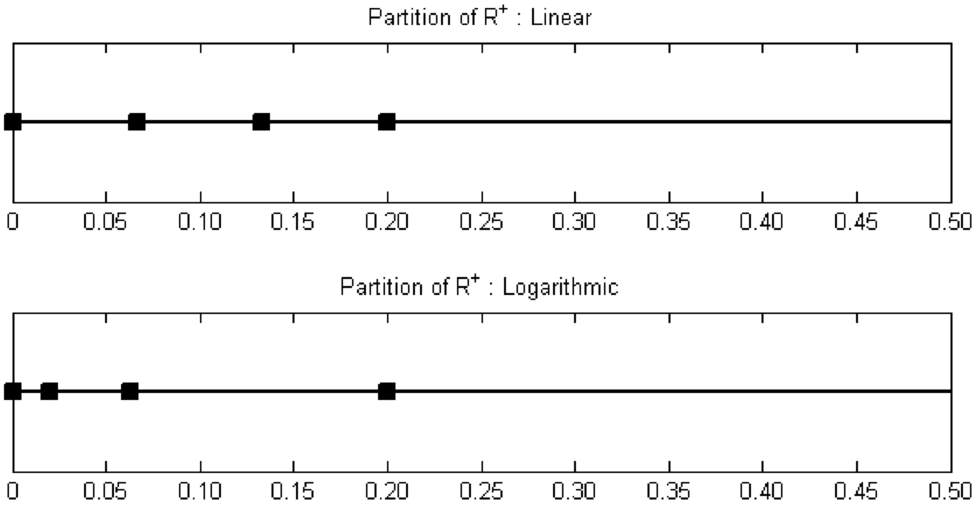


Figure 13: Edges of intervals composing partitions: (a) linear, (b) logarithmic

In a linear partition of $\mathbb{R}^+$, the edges of the intervals composing the partition are linearly spaced between 0 and certain value, and the last interval ranges from this value to infinity. Mathematically, a linear partition of $\mathbb{R}^+$, $\{T_i\}$, containing h intervals (h is a natural number), is the following:

$$T_i = \left\{t \in \left[(i-1)\cdot\frac{t_{max}}{h-1}, i\cdot\frac{t_{max}}{h-1}\right)\right\}; i = 1, \dots, h-1$$

$$T_h = \{t \in [t_{max}, \infty)\}$$

Where $t_{max} \in \mathbb{R}^+$ and $h \in \mathbb{N}$.

In a logarithmic partition of $\mathbb{R}^+$, the edges of the intervals composing the partition are logarithmically spaced until a certain value, and last interval ranges from this value to infinity. Mathematically, a logarithmic partition of $\mathbb{R}^+$, $\{T_i\}$, containing h intervals (h is a natural number), is the following:

$$T_1 = \{t \in [0, t_{min})\}$$

$$T_i = \left\{ t \in \left[10^{\log(t_{min})+(i-2)\cdot\frac{\log(t_{max})-\log(t_{min})}{h-1}}, 10^{\log(t_{min})+(i-1)\cdot\frac{\log(t_{max})-\log(t_{min})}{h-1}} \right) \right\};$$

$$i = 2, \dots, h-1$$

$$T_h = \{ t \in [t_{max}, \infty) \}$$

Where $t_{min}, t_{max} \in \mathbb{R}^+$ and $h \in \mathbb{N}$. t_{min} will be a value close to 0.

Figure 13 represents the edges of the intervals composing two possible partitions of $\mathbb{R}^+$. In the first one, the edges are linearly spaced between 0 and a value $t_{max} = 0.2$. All the intervals have the same length, excluded the last one, which goes from the maximum established value to infinity. In the second partition, the first interval would range from 0 to $t_{min} = 0.02$; sucesive intervals would have logarithmically spaced edges, between 0 and $t_{max} = 0.2$; and the last interval would range from this value to infinity.

To find a suitable value for the lower edge of t_{max}, we have studied the distribution of values of $IAT[n]$. In Figure 11 we can see that the probability of having an interarrival time higher than 0.0348 is 10%, and the probability of having an interarrival time higher than 0.0775 is 2%. Very few packets will arrive with higher interarrival times, so it makes no sense to continue dividing the space above these values.

4. Results

This chapter presents descriptive statistics of the set of packet traces used to evaluate the methods presented in chapter 3, as well as the results of the evaluations of the prediction system described in section 3.3 and the method of estimation of upper bounds for interarrival time described in 3.4.

4.1. Set of packet traces

For this evaluation we have a set of traces of 12 million user data packets, captured at the Gi interface of a commercial UMTS network. Each packet has a timestamp, length, direction, and owner. The owner is represented as a temporary subscriber identification number. This allows us to identify the traffic of each UE, without violating the privacy of the users of the network. Traces also include information about the operating system (OS) running in the UE which generated the trace.

This set of traces is the result of capturing the traffic passing through the Gi interface during one hour (3600 seconds). One filter was applied to the capture: only packets whose 'Operating System' attribute is Apple's iOS are included. The capture of traffic was made at the busy hour of one day in June 2010.

A first analysis of the temporary subscriber identification number of the traces shows that there are 5212 different values of this field through the complete set of traces. Note that this does not imply the presence of 5212 *different* UEs, since temporary IDs can change over time. However, it is granted that packets with the same temporary identification number are sent or received by the same UE. The complete set will be divided into this number of subsets, where each subset contains traces of packets to or from a single UE.

The traces do not include information about the connectivity state of the UE or the configuration of the RNCs managing the states of the UEs as the data was collected at the Gi rather than Iub. For this reason assumptions will need to be made.

4.1.1. Self-similarity

We show evidences of self-similarity of the aggregate traffic, using methods and arguments presented in [8].

Figure 14 shows the number of bytes transferred through the Gi interface within a certain time unit, for four different time units. The plots show the similarity of the traffic in four different time scales.

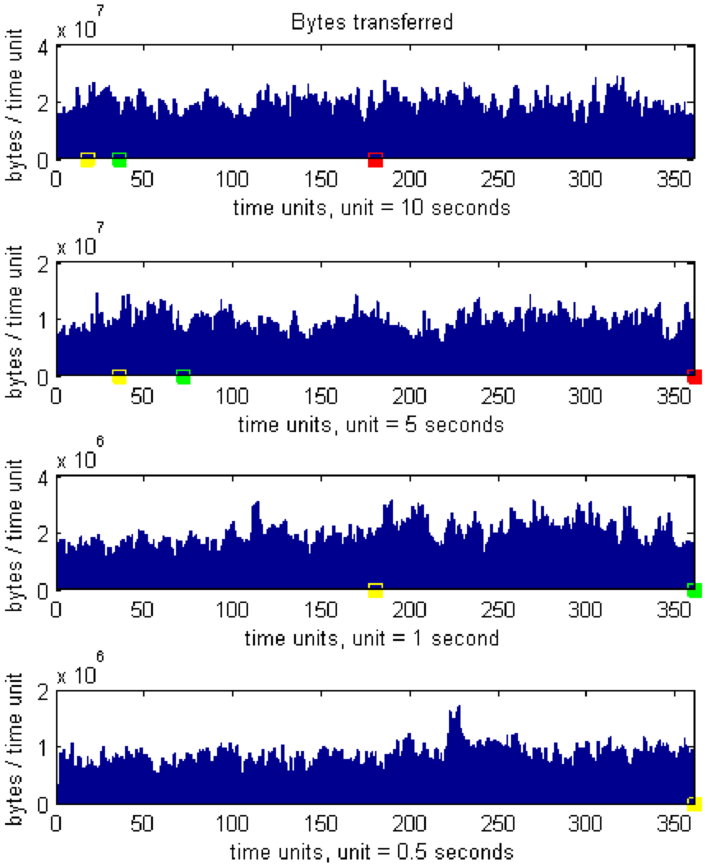


Figure 14: Pictorial "proof" of self similarity.

We analyze the correlation of the traffic for different levels of aggregation, considering the sequence of transferred bytes in time units of 0.1 seconds. From this sequence, we calculate m-aggregated processes, as defined in section 2.2.2.2. Figure 15 shows the autocorrelation of 360 samples of three m-aggregated processes.

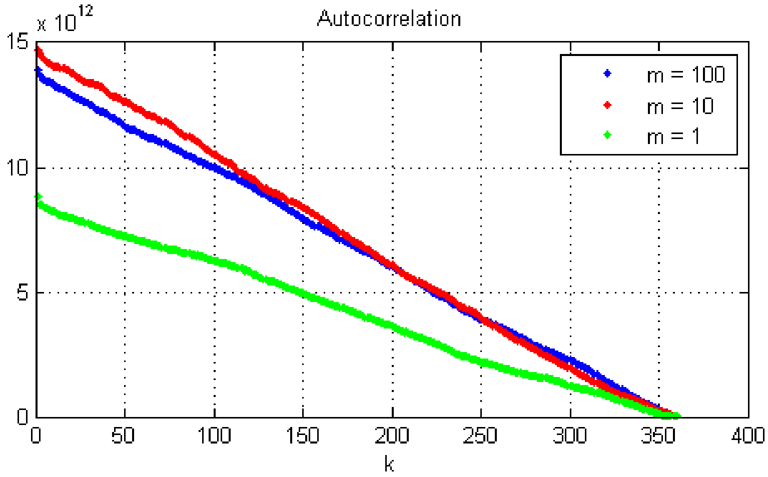


Figure 15: Autocorrelation of 360 samples of three m-aggregated processes

Figure 15 shows that for large values of m and k the sequences of autocorrelation are similar, which according to [8] is an evidence of asymptotical self-similarity.

We use one of the methods presented in [8] to make an estimation of the Hurst parameter, H, introduced in section 2.2.2.1. Figure 16 shows the variance of different m-aggregated processes, in a log-log scale, called a variance-time plot. According to [8], a value of asymptotic slope of this plot, in logarithmic scale, between -1 and 0, suggest self-similarity. Figure 16 shows the asymptotic slope, estimated from the resulting points of the variance-time plot for large values of m (100, 200, and 500) using least squares fitting. The slope is estimated to be -0.34, quite different from -1 (shown in red).

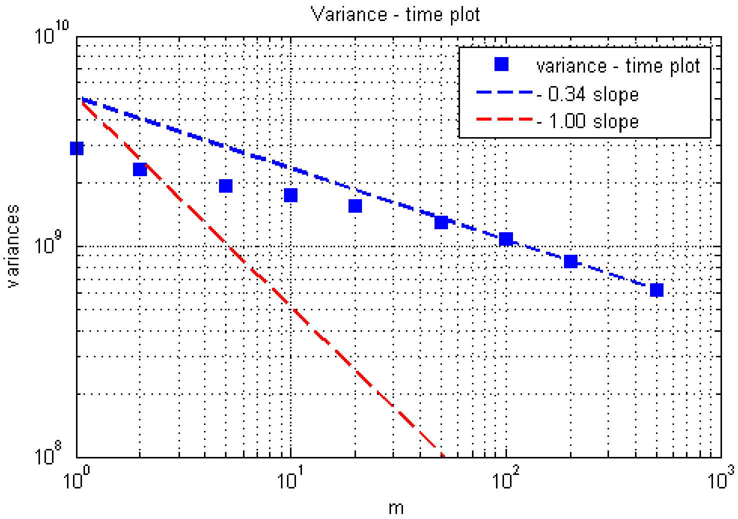


Figure 16: Variance-time plot

With this estimated value of the asymptotical slope, we get an estimation of the Hurst parameter:

$$H \approx 1 - \frac{0.34}{2} = 0.83$$

Figure 17 shows an estimation of the power spectral density of the sequence of transferred bytes in time units of 0.1 seconds. The low frequency part is characteristic for a power law behavior of the spectral density around zero. According to [8], this behavior is observed in self similar processes.

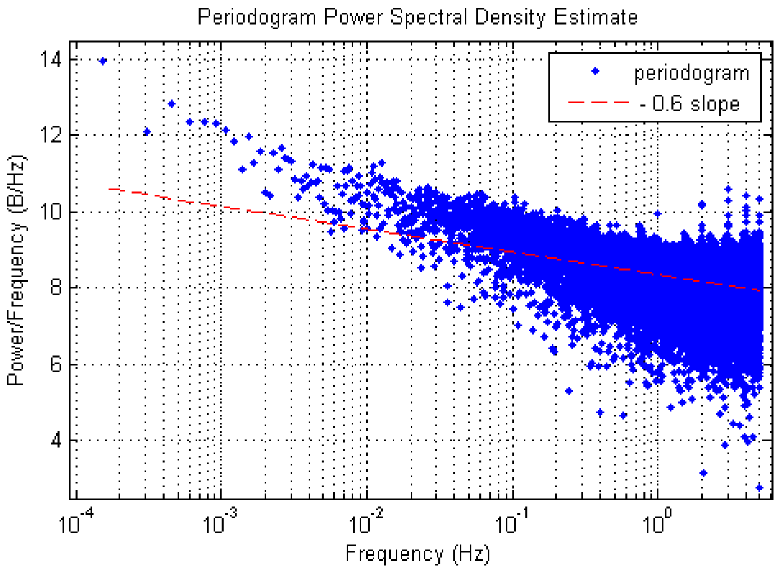


Figure 17: Periodogram of the aggregate traffic

Another method proposed in [8] for the estimation of the Hurst parameter, H, is based in the slope of the line obtained by a least squares fitting of the periodogram, expressed in bels (B) rather than in decibels (dB). The slope we obtain in our traffic is -0.6. Applying the method of estimation of H:

$$H \approx \frac{1 + 0.6}{2} = 0.8$$

The Hurst parameter is a measure of the degree of self similarity of a process (see section 2.2.2.2). The Hurst parameter is a value between 0.5 and 1: the closer it is to 1, the higher the degree of self similarity is. Both methods give a similar value of the Hurst parameter, close to 0.8. This suggests that the aggregate traffic of the UEs is indeed a self-similar process. Consequently, it will present long range dependency (see section 2.2.2.1).

4.1.2. Length of packets

Our set of packet traces includes 12 millions of packets: 46.22% of them are uplink packets, 53.78% are downlink packets. Figure 18 shows histogram-based estimations of the distributions of the length of the packets. The distribution of the length of uplink packets shows a peak at 52 bytes and for downlink packets a peak at 1450 bytes. The length of 1450 represents the largest size packets that can pass through the core network without fragmentation. With roughly 80% of downlink packets having this size, this represent an aggregate downlink data rate of about 17 Mbps (for the hour).

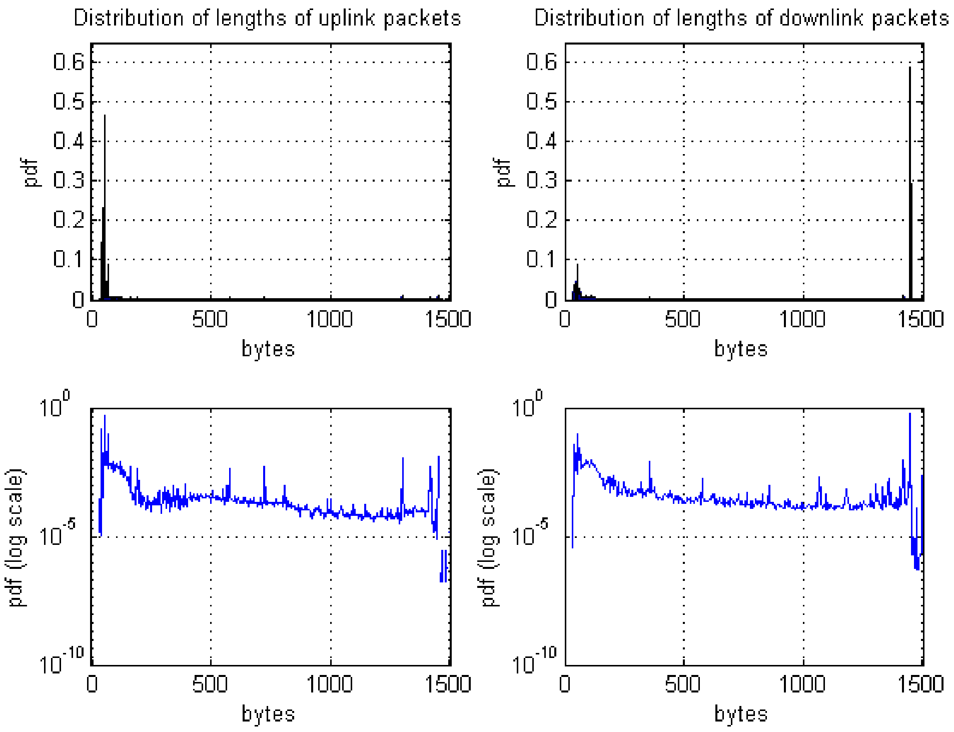


Figure 18: Distributions of lengths of uplink (left) and downlink (right) packets

The whole set will be divided into 5212 subsets, where each subset contains traces of packets to or from a single UE. Figure 19 shows the distribution of the number of packets per subset. 84% of the subsets contain less than 2000 packets. These subsets contain a 12.64% of the whole number of packets. This means that the major part of the packets is contained within 834 subsets, and each of these subsets contains a large number of packets each.

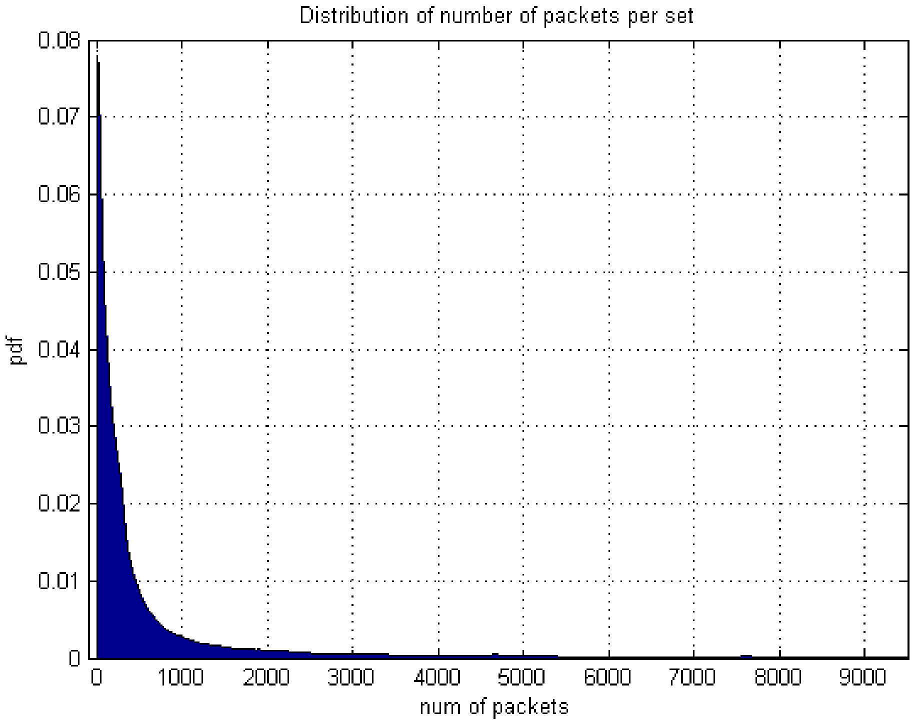


Figure 19: Distribution of the number of packets per set

4.1.3. Interarrival time of packets

We analyze the interarrival time of packets within the same subset. Figure 20 shows histogram-based estimations of the distributions of interarrival time. For each packet, the interrarrival time is the time between the arrival of the immediately previous packet in its subset and in this same direction, and the arrival of this packet.

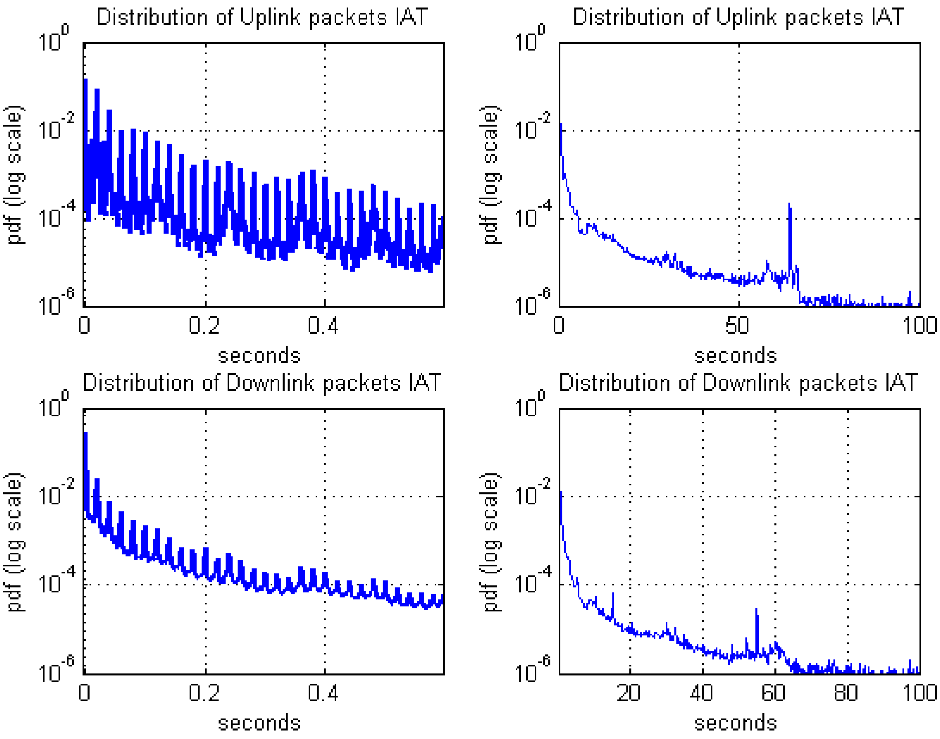


Figure 20: Distributions of interarrival time of uplink and downlink packets

Since the packets are captured at the Gi interface, arrivals of uplink traffic are affected by parameters of the UMTS networks (channel allocation algorithms, timeouts, etc.), while arrivals of downlink packets are primarily affected by the source of the packets and the network between the source and the Gi interface. The distribution of interarrival time of uplink packets shows peaks at multiples of 20 milliseconds. These peaks may be due to the structure of the physical layer of the interface between UE and RNC. A physical channel is a certain time slot at a certain carrier frequency. These time slots compose radio frames. The structure of a frame is repeated in time periodically. A UE transmitting through a dedicated channel transmits in the same slot in every frame, so this communication between the UE and the RNC takes place periodically. Figure 20 suggests that this period is 20 ms. See [12] and [13] for more details. Note that it is unlikely that the uplink user is allocated a high speed channel as the aggregate average data rate (assuming 52 byte packets for the majority of uplink traffic) is only 640,917 bps. Since most of this is due to the 834 most active users, this corresponds to less than 770 bps per user (averaged over the hour).

Downlink traffic is not affected directly by parameters of the UMTS network. But it is correlated with uplink traffic, so these parameters also indirectly affect downlink traffic. For this reason, we can see smoother peaks in the distribution of interarrival times of downlink packets, but this traffic also shows peaks at multiples of 20 ms.

4.2. Presence of patterns

Rows of the table used to estimate p_A, described in section 3.3.4, have a one to one correspondence with the set of all the possible sequences and subsequences of events stored in the buffer. For each row, the value of the sum of the three columns, $m_{UL,k}$, $m_{DL,k}$ and $m_{I,k}$, is computed using the process described in section 3.3.7 for the traffic of a UE. This value is the number of times the sequence of events corresponding to the row was detected in the traffic of this UE. The value of this sum will be higher for those rows corresponding to the sequences of events most frequently repeated in time.

Every subset of packet traces for an individual UE was processed as described in section 3.3.6. Here we only consider those traces in which more than 200 events are observed. This means that we considered 956 traces, containing 8193079 packets out of 12000000, which represents a 68.28% of the total number of packets. We used the following parameters:

Packet burst threshold: $\tau_{PB} = 0.5$ seconds.
Application burst threshold: set statically to 2 seconds, for all users. $\tau_{AB} = 2$ seconds.
Size of buffer: $n = 10$ events.

After processing every subset, the sum of the values $m_{UL,k}$, $m_{DL,k}$ and $m_{I,k}$ is calculated for every row of the table. Remember that when an inactivity event is detected, previous events in the buffer are discarded. This means that sequences of fewer events have more occurrences registered in the table than sequences with a greater number of events. We normalize the number of occurrences of each sequence to compensate for this fact. The normalization factor for each row is $\frac{n_k}{n}$, where n_k is the number of events of the sequence corresponding to row k of the table. This normalization allows us to compare the number of occurrences of sequences of different numbers of events. We calculate the fraction of occurrences of each sequence from the 'normalized' number of occurrences of each sequence. This value will be comparable between subsets of traces containing different numbers of events. If all the possible sequences were uniformly distributed, then all of them would present the same fraction of occurrences, and this value would be equal to $\frac{1}{r}$, where r is the number of rows in the table. This is the average value of the fraction of occurrences of all the possible sequences. The variance of these values would be 0 (since they would be uniformly distributed).

However, it is observed that, within a subset, sequences are not uniformly distributed: some sequences tend to occur more times, while others are never detected. Hence, the fraction of occurrences is higher for some sequences in this subset. Since the sum of fractions of occurrences of sequences within every subset is 1, the fraction of occurrences of every sequence will move away from the average value, $\frac{1}{r}$. Consequently, within a subset, the presence of sequences of events which tend to be repeated in time will be reflected in the variance of all the fractions of occurrences of the subset. The more frequent one or more sequences of events are repeated in time, the higher the value of the variance of the fractions of occurrences will be.

We calculated the variance of the fractions of occurrences of all the sequences for every subset of traces processed. Figure 21 shows the distribution of this parameter among the subsets. The average value of this distribution is $0.44805 \cdot 10^{-5}$. The plot shows the presence of subsets with a high variance of the fractions of occurrences: for 8.13 % of the subsets this parameter is greater than 10^{-5}.

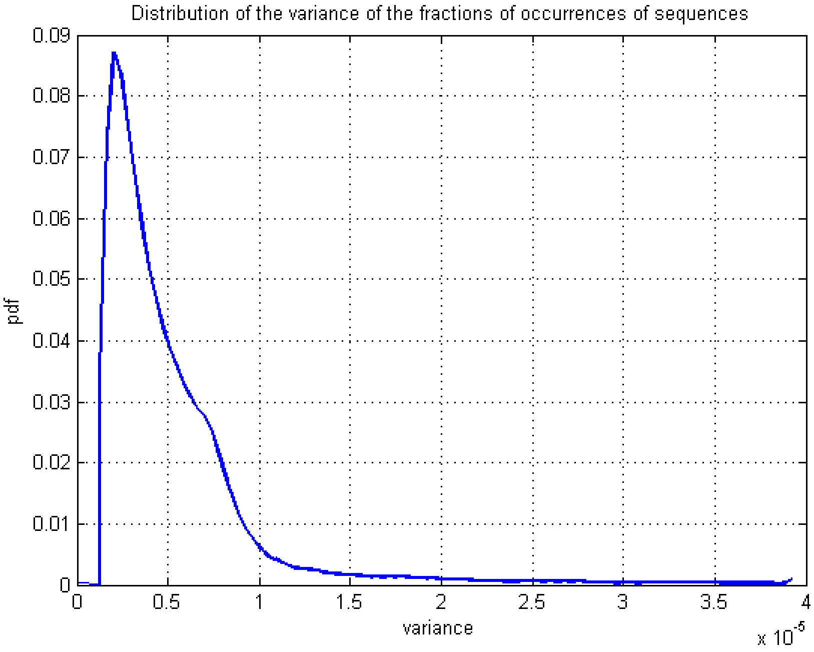


Figure 21: Distribution of the variance of the fractions of occurrences

We consider that the most repeated sequence in a subset of traces is the one which presents the highest ‘normalized’ number of occurrences in the subset. For every subset of traces we observe the number of events composing the most repeated sequence. Figure 22 show how the length of the most repeated sequence is distributed among the considered subsets of traces.

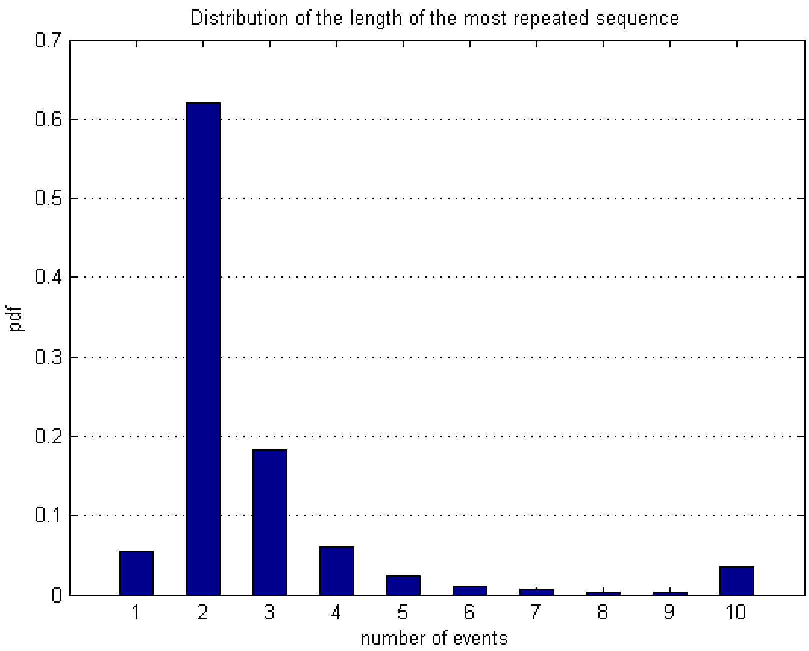


Figure 22: Distribution of the length of the most repeated sequence.

We have observed that the most common sequence is uplink packet burst, followed by downlink packet burst. This suggests that the traffic corresponds to applications with a client-server architecture, for example, web browsing. UEs send requests (uplink packet burst) and in a short time they get a response from the server (downlink packet burst).

4.3. Evaluation of prediction system

This section presents the results of the evaluation of the prediction system described in section 3.3. This evaluation is based on the parameters introduced in subsection 3.3.6: reduction of inactivity time of the UE while it is in Cell_DCH state, cost paid in terms of reduction of the packet burst throughput, and rate of correct predictions.

4.3.1. Reference system

To estimate the reduction in the inactivity time and burst throughput caused by the prediction system described in section 3.3, we take as a reference a system where the transmission or reception of a packet burst requires that the UE is Cell_DCH state. After the transmission or reception of every packet burst, the UE is kept in this state for a time, τ_{AB}, to avoid subsequent packet bursts coming next suffering a delay due to the state switching of the UE. After this time, if there is no transmission or reception of a packet burst, then the UE will transition to a state of lower power consumption. This is equivalent to a system with predictions, where the probability of activity, p_A, is taken as 1 at every prediction moment.

Every subset of packet traces for an individual UE was processed as described in section 3.3.6. For this analysis, we only considered those traces in which more than 200 events are observed. This means that we considered 956 traces, containing a total of 8193079 packets out of 12000000, which represents a 68.28% of the total number of packets. We will take the following values for the required parameters:

Packet burst threshold: $\tau_{PB} = 0.5$ seconds.

Application burst threshold: set statically to 2 seconds, for all users. $\tau_{AB} = 2$ seconds.

Since in this reference system $p_A = 1$, regardless of the previous events, there will be no need to specify a size for the event buffer. For each subset of the packet traces the total inactivity time, $t_{IA,ref}$, is calculated as the sum of all partial inactivity times at every prediction moment (see section 3.3.6).

The mean value of the burst throughput, $E[R_{PB}]_{ref}$, is calculated considering all the detected packet bursts. We will make assumptions about the bit rate and the delay due to the signaling for state switching. We describe these assumptions below.

The bit rate is not a constant parameter, but rather it depends on the connectivity quality where the UE is. We consider three possible bit rates, whose values are shown in Table 2. Packet traces will be processed for each case, and within each case the bit rate will be considered invariant.

Table 2: Values of bit rates

	Worst case	Typical case	Best case
Uplink bit rate	20 Kbps	200 Kbps	2 Mbps
Downlink bit rate	100 Kbps	1 Mbps	10 Mbps

The delay due to signaling for state switching (t_{switch}) will be considered to be 0.5 seconds if the burst is an uplink burst, and 2 seconds if it is a downlink burst.

We take this system as a reference because it is similar to the mechanisms currently used in UMTS networks. This will allow us to evaluate the potential benefits of the prediction system.

4.3.2. Static τ_{AB}

The process is repeated for every subset of packet traces, assuming the same numerical values, but calculating values for p_A as described in section 3.3.4. The value of τ_{AB} will be statically set to 2 seconds, as in the reference system.

For every subset of traces in which more than 200 events are observed, we calculate the fraction of correct predictions. Figure 23 shows how this parameter is distributed among the 956 subsets, assuming three different values of buffer size, n.

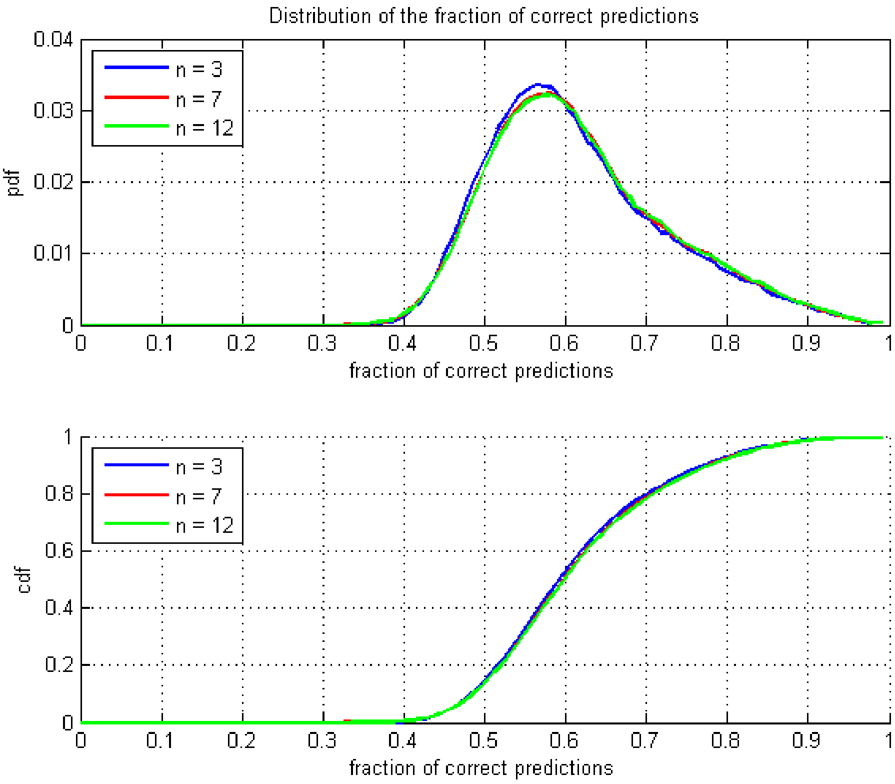


Figure 23: Distribution of the fraction of correct predictions, static τ_{AB}.

Figure 23 shows that for most of the subsets the fraction of correct predictions is over 0.5 (which would be a random binary prediction). The minimum observed fraction of correct predictions of a subset of traces is 0.33, while the maximum is near 1. The most frequent value is 0.56, with $n = 3$, and 0.58, with $n = 7$ and $n = 12$.

For every subset of traces in which more than 200 events are observed, we quantify the reduction of the inactivity time and the burst throughput, assuming different sizes of the event buffer (n). See section 3.3.6 for definitions of the parameters t_{IA} and $E[R_{PB}]$.

Reduction of inactivity time: $\Delta t_{IA} = 1 - \frac{t_{IA,n}}{t_{IA,ref}}$

Reduction of the burst throughput: $\Delta E[R_{PB}] = 1 - \frac{E[R_{PB}]_n}{E[R_{PB}]_{ref}}$

$t_{IA,ref}$ and $E[R_{PB}]_{ref}$ are the values obtained from the reference system described in section 4.3.1.

Figure 24 shows the average of the values of Δt_{IA} obtained for the 956 subsets of traces, for different values of n. Figure 25 shows the average of the obtained values of $\Delta E[R_{PB}]$, considering the values shown in Table 2 for three cases of bit-rate.

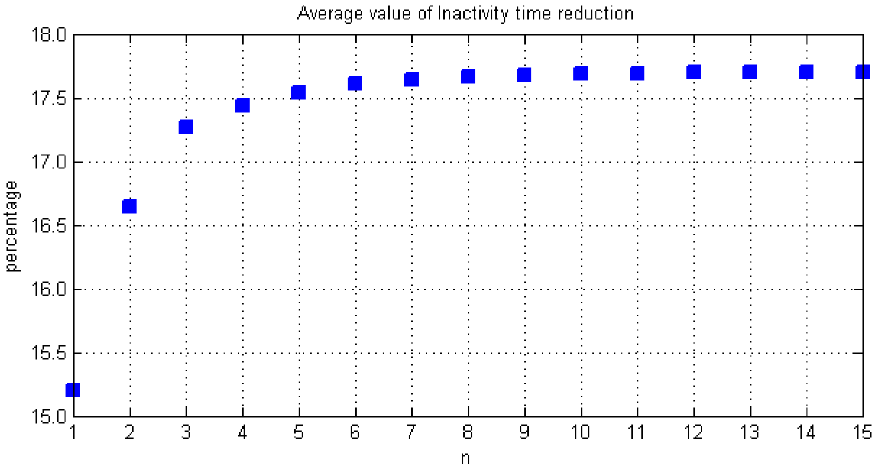


Figure 24: Average value of inactivity time reduction, static τ_{AB}

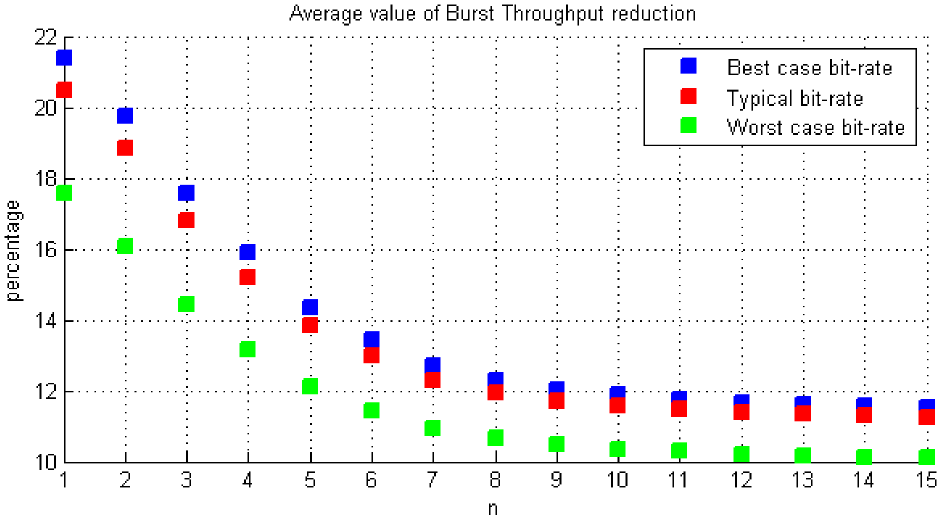


Figure 25: Average value of burst throughput reduction, static τ_{AB}

Figure 24 shows that the greater the number of previous events considered to make the predictions is, the larger the inactivity time reduction is obtained. Figure 25 shows that the burst throughput is reduced a lot when we use small values of n (18% with $n = 1$), while it is only reduced by about 10% with $n = 15$. This means that the penalty for guessing that the link should be taken down in the next instant decreases when we use a large n for the prediction.

Figure 26 and Figure 27 show how the parameters are distributed among the considered subsets of traces.

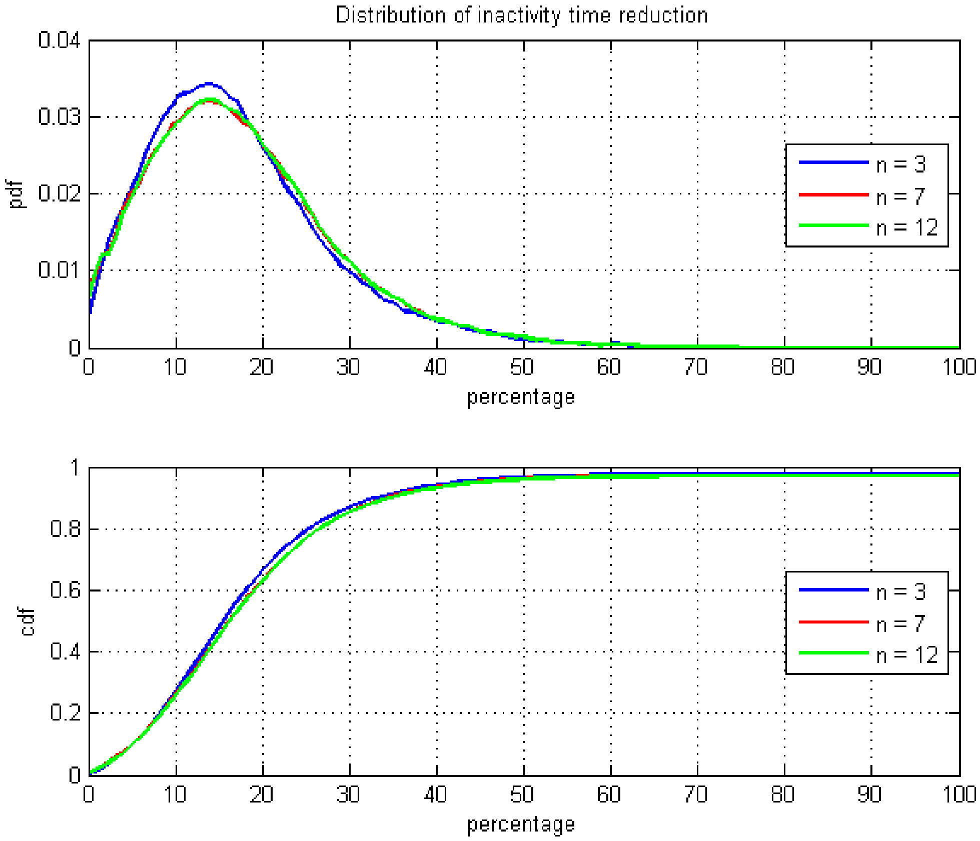


Figure 26: Distribution of inactivity time reduction, static τ_{AB}

The most frequent value of inactivity time reduction is 13.7%. Few subsets get a reduction greater than the 50%. Figure 26 also shows the presence of subsets which do not get any reduction of their inactivity time.

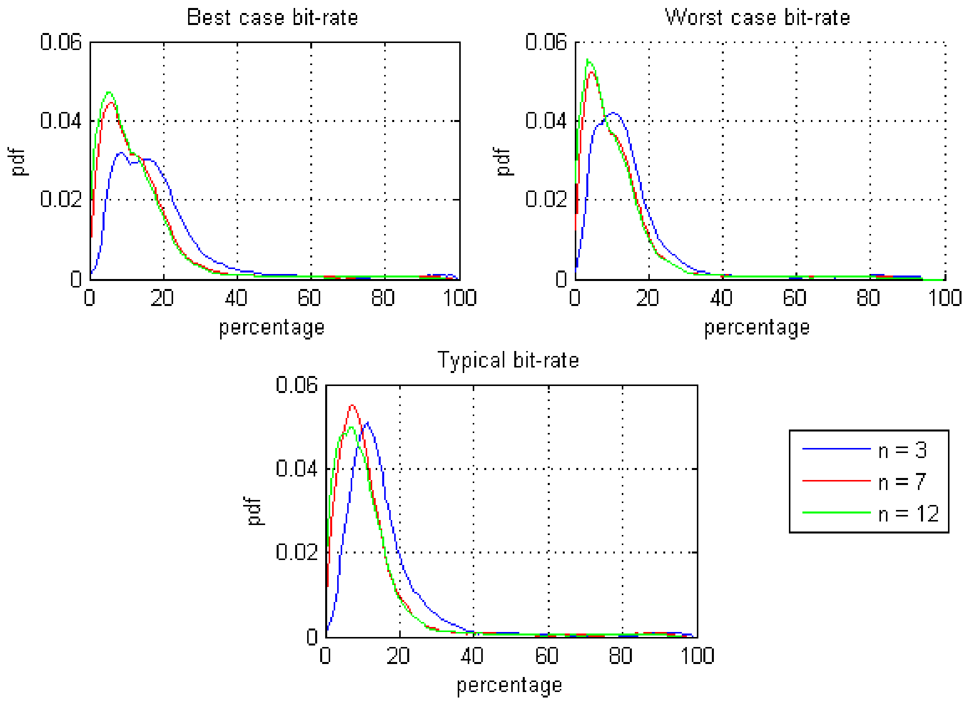


Figure 27: Distribution of burst throughput reduction, static τ_{AB}

The most frequent value of throughput reduction is between 11% and 12% for $n = 3$, between 5% and 7% for $n = 7$, and between 4% and 6% for $n = 12$. For most subsets, the reduction in their burst throughput is not greater than 20% in any case.

4.3.3. Dynamic τ_{AB}

Now we evaluate a system where the value of τ_{AB} is calculated dynamically for every UE at every prediction moment, as described in section 3.3.5. Values for p_A are calculated as described in section 3.3.4. We will continue to use the reference system described in section 4.3.1.

The parameters used for the dynamic calculation of τ_{AB} are:

$IAT_{PB,max} = 8$ seconds. This means that interarrival times of packet bursts higher than 8 seconds will not be considered when calculatingτ_{AB}, and that when an interarrival time higher than 8 seconds is detected, the previously stored values of interarrival times of packet bursts will be discarded.

$\tau_{AB,def} = 1.25$ seconds. When there is no information about previous interarrival times of packet bursts, τ_{AB} will be set to this value.

$h = 2$. The value of τ_{AB} will be calculated as the average value of the two last stored values of interarrival times of packet bursts.

For every subset of traces in which more than 200 events are observed, we calculate the fraction of correct predictions. Figure 28 shows how this parameter is distributed among the 956 considered subsets, assuming three different values of size of buffer, n.

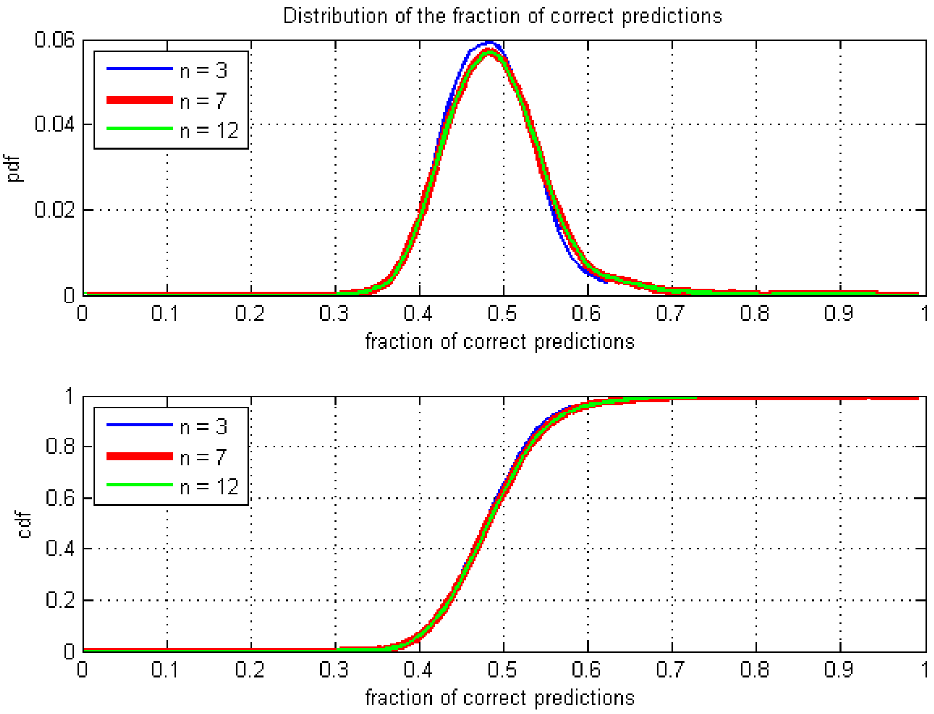


Figure 28: Distribution of the fraction of correct predictions, dynamtic τ_{AB}.

Figure 28 shows that in this system there are fewer subsets whose fraction of correct predictions is over 0.5. The most frequent value is 0.48 for the three considered values of n. The minimum observed fraction of correct predictions of a subset of traces is 0.31. With this method there are fewer correct predictions. The gains of using this method come from the dynamic calculation of τ_{AB}, rather than from the correct predictions.

Figure 29 shows the average of the values of the inactivity time reduction, Δt_{IA}, obtained for the 956 subsets of traces, for different values of n. Figure 30 shows the average of the obtained values of the burst throughput reduction, $\Delta E[R_{PB}]$, considering the values shown in Table 2 for three cases of bit-rate.

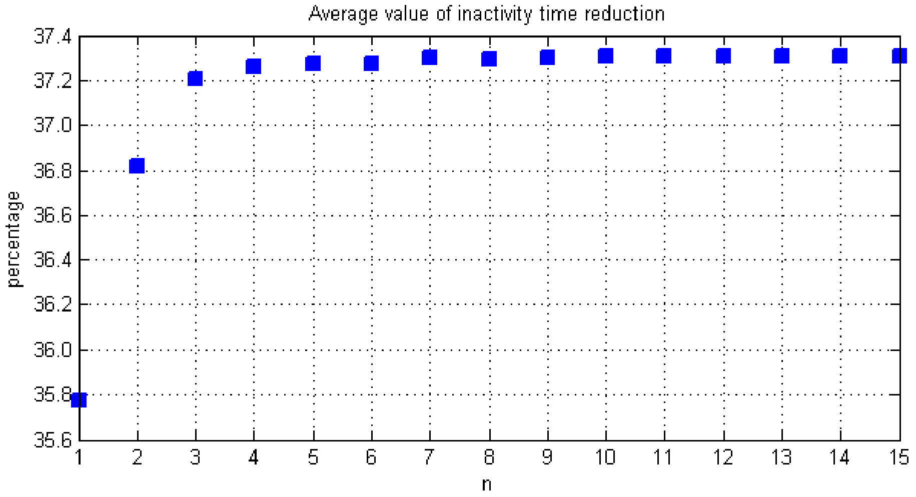


Figure 29: Average value of inactivity time reduction for different values of n, dynamic τ_{AB}

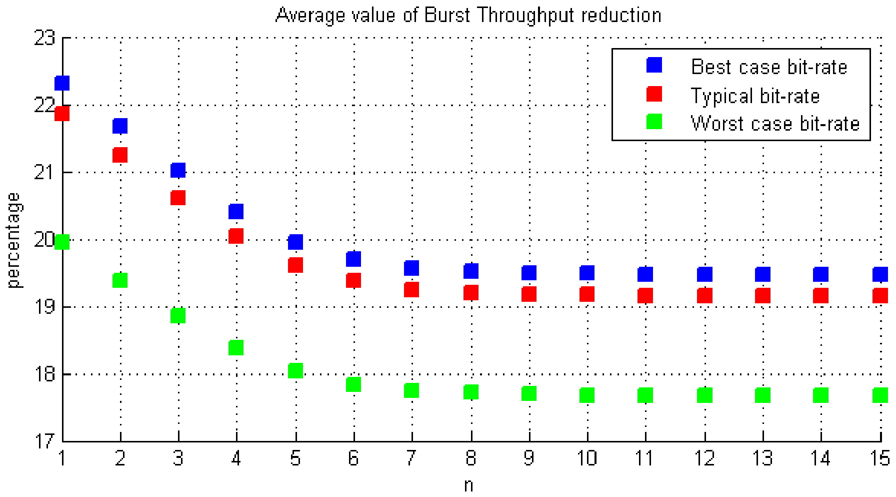


Figure 30: Average value of burst throughput reduction for different values of n, dynamic τ_{AB}

Figure 29 and Figure 30 show that the greater the number of previous events considered to make the predictions is, the better values of inactivity time reduction and burst throughput reduction are obtained. For values of n greater than or equal to 9, the reduction of inactivity time and the reduction of the burst throughput remain constant, at values which are higher than the values obtained in the system with static τ_{AB} (Figure 24 and Figure 25). This means that a higher reduction of inactivity time is achieved, but a higher cost is paid in terms of burst throughput reduction.

Figure 31 and Figure 32 show how the parameters are distributed among the considered subsets of traces.

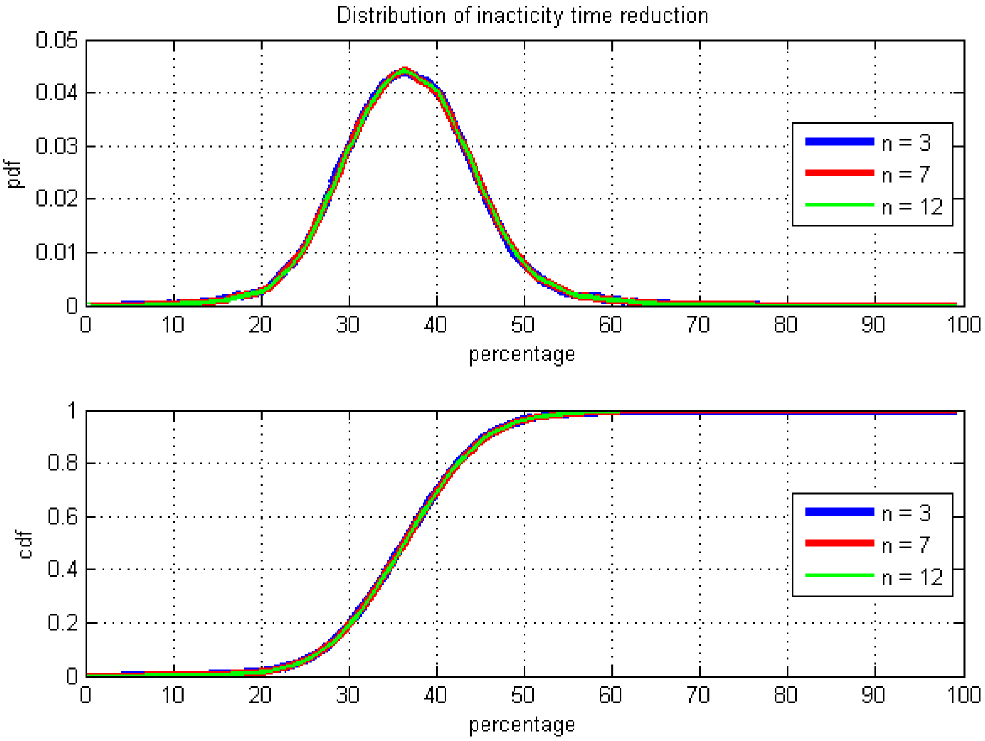


Figure 31: Distribution of inactivity time reduction for different values of n, dynamic τ_{AB}

The most frequent value of inactivity time reduction is 36.3%, which is greater in the system with static τ_{AB}. Few subsets get a reduction greater than the 50%, but, unlike the system with static τ_{AB} (see Figure 26), in this system all the subsets get some reduction of their inactivity time in Cell_DCH state.

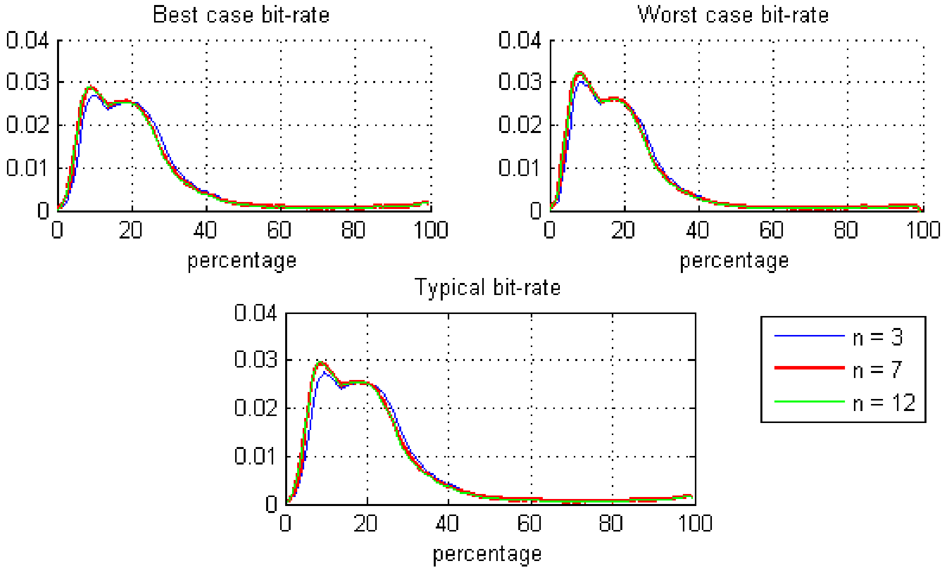


Figure 32: Distribution of burst throughput reduction for different values of n, dynamic τ_{AB}

Figure 32 shows a wide dispersion of throughput reductions. This value is between 10% and 20% for most of the subsets. For most of the subsets, the reduction in burst throughput is less than 40% in any case. This upper bound is higher than observed in the system with static τ_{AB}.

We also analyze how the parameter h affects the inactivity time reduction and the burst throughput reduction. Remember that h is the maximum number of previous interarrival times of packet bursts stored in order to calculate their average in order to set the value of τ_{AB}. The parameters used this analysis are:

Size of the event buffer, $n = 15$. This means that predictions of next events will be made considering the previous 15 events.

$IAT_{PB,max} = 8$ seconds. This means that interarrival times of packet bursts longer than 8 seconds will not be considered when calculating τ_{AB}, and that when an interarrival time higher than 8 seconds is detected, the previously stored values of interarrival times of packet bursts will be discarded.

$\tau_{AB,def} = 1.25$ seconds. When there is no information about previous interarrival times of packet bursts, τ_{AB} will be set to this value.

Figure 33 shows the average of the values of the inactivity time reduction, Δt_{IA}, obtained for the 956 subsets of traces, for different values of n. Figure 34 shows the average of the obtained values of the burst throughput reduction, $\Delta E[R_{PB}]$, the bit-rate shown in Table 2 for a typical situation.

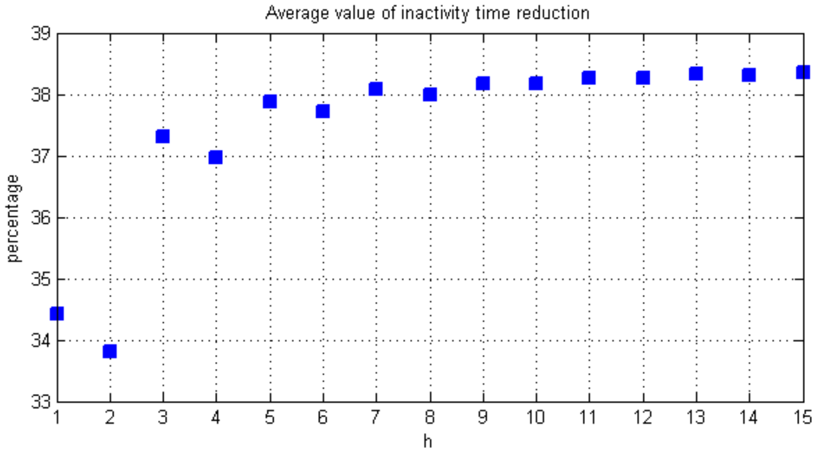


Figure 33: Average value of inactivity time reduction for different values of h, dynamic τ_{AB}

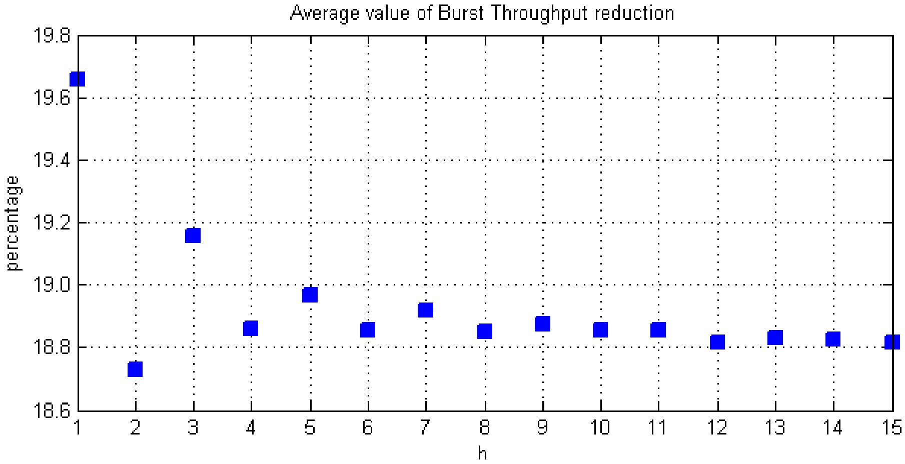


Figure 34: Average value of burst throughput reduction for different values of h, dynamic τ_{AB}

Figure 33 and Figure 34 show that for values of *h* from 12, the average value of the inactivity time reduction is stabilized at 38.3%, and the value of the burst throughput reduction is stabilized at 18.83%.

Figure 31 and Figure 32 show how these parameters are distributed among the considered subsets of traces.

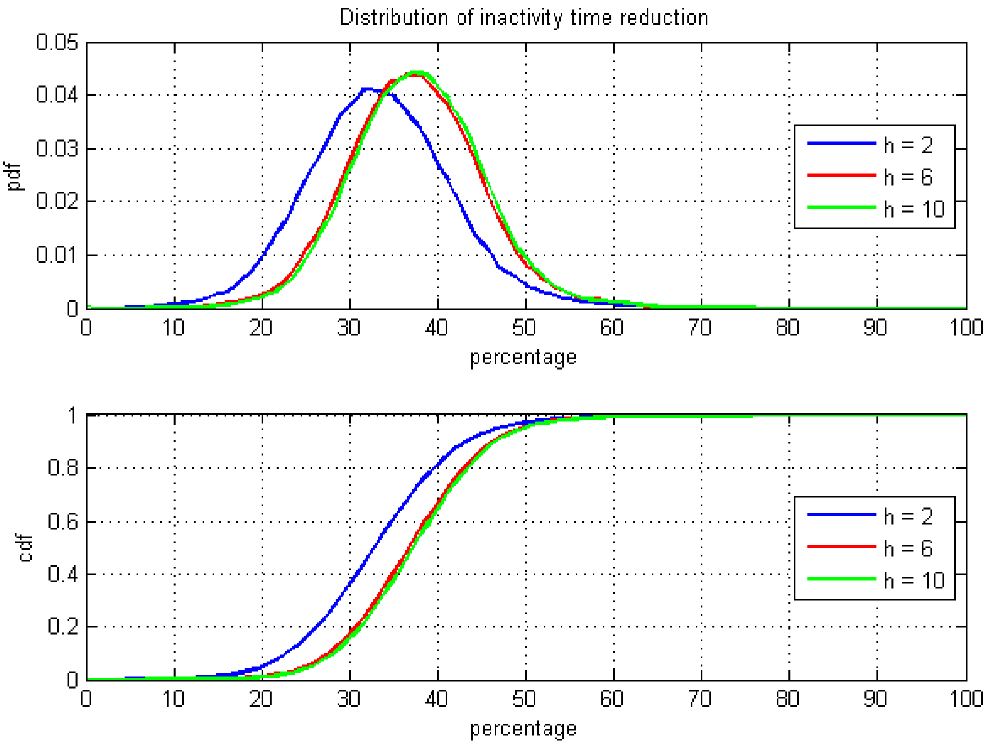


Figure 35: Distribution of inactivity time reduction for different values of h, dynamic τ_{AB}

The most frequent value of inactivity time reduction is 32.2% for $h = 2$, and 37% for $h = 6$ and $h = 10$. Few subsets get a reduction greater than the 50%, but for $h = 6$ and $h = 10$ more subsets get this reduction.

Figure 36 shows the most frequent value of burst throughput reduction is 13%. For most of the subsets, the reduction in their burst throughput is less than 40% in any case.

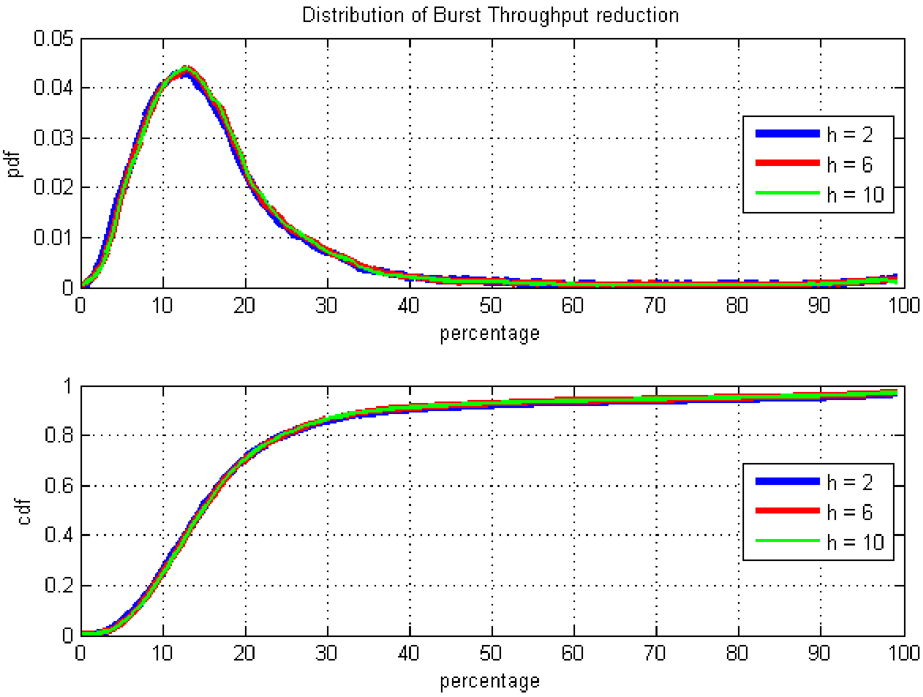


Figure 36: Distribution of burst throughput reduction for different values of h, dynamic τ_{AB}

4.4. Estimation of upper bounds for IAT

This section presents an evaluation of the method described in section 3.4 to estimate an upper bound for the interarrival time of the packets.

4.4.1. Different values for m

The value of m is the maximum number of previous samples of $IAT[n]$ to be considered when estimating the upper bound of $IAT[n+1]$. Using a value of $k > 0$, all these values meet a condition: their previous k values are in the same ranges in which the values $IAT[n-k+1], \ldots, IAT[n-1], IAT[n]$ are.

For this analysis we use the following values for the different parameters (see section 3.4.2 for details):

$k = 3$. The condition of the samples considered to estimate the upper bound will involve their previous three interarrival times.

$p = 0.9$. This means that the estimated upper bound of the interarrival time of the next packet for every packet will be the quantile of the distribution of $IAT[n]$ conditioned to the previous $k = 3$ samples with a probability of 0.9. Ideally, 90% of the packets will have an interarrival time which will be below their estimated upper bound.

The partition of $\mathbb{R}^+$ used to classify the k previous values of $IAT[n]$ and to estimate the conditional distribution according to this classification will have $h = 10$ intervals. Edges will be logarithmically spaced between $t_{min} = 10^{-6}$ and $t_{max} = 0.1$ seconds.

For every subset of traces to/from the same UE, containing at least 5000 packets[3], we evaluate the percentage of packets whose interarrival time is below their estimated upper bound, and the average value of all the estimated upper bounds. 438 subsets were analyzed, containing 9 millions of packets. This is the 75% of the packets included in the whole set.

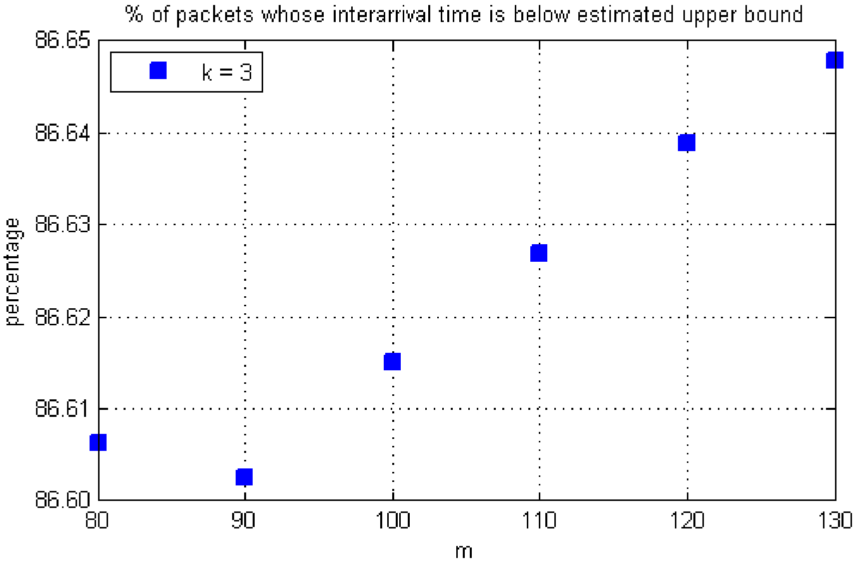


Figure 37: Percentage of packets whose interarrival time is below its estimated upper bound, for different values of m

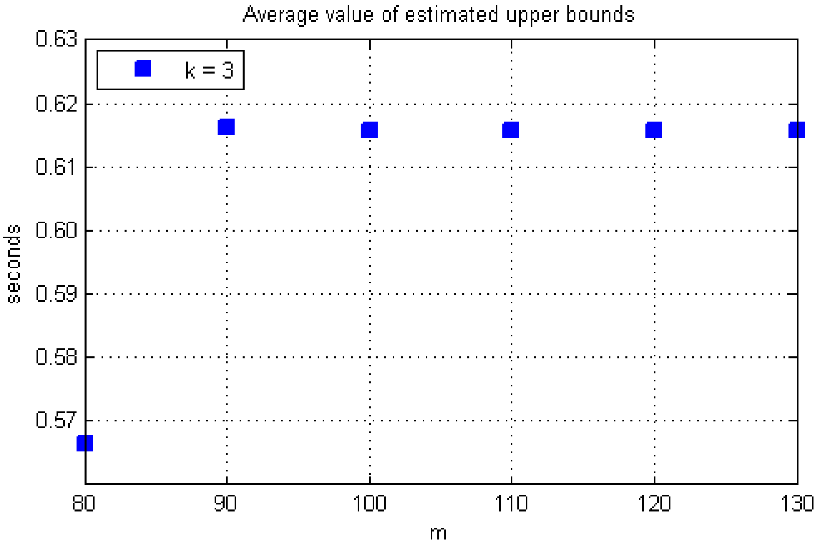


Figure 38: Average value of estimated upper bounds, for different values of m

Increasing m means increasing the number of samples from which the conditional distributions are estimated, making estimations more accurate. This causes that more packets have their interarrival time below the estimated upper bound (Figure 37). However, Figure 38 shows the average value of the upper bounds, remains constant for higher values of m.

[3] Note that this method is based in arrivals of packets, rather than events. This is the reason why we choose a different set of subsets for the analysis, applying a selection criteria based in the number of packets contained in the subset, not in the number of events contained.

4.4.2. Different values for k

The parameter k is the number of previous packets whose interarrival time is considered in order to estimate the conditional distributions of the interarrival time of the next packet, and an upper bound for the interarrival time of the next packet based on the conditional distribution.

For this analysis we use the following values for the different parameters (see section 3.4.2 for details):

$p = 0.9$. This means that the estimated upper bound of the interarrival time of the next packet for every packet will be the quantile of the distribution of $IAT[n]$ conditioned to the previous k samples for a probability of 0.9. Ideally, 90% of the packets will have an interarrival time which will be below their estimated upper bound.

$m = 100$. Conditional distributions will be estimated taking, at most, 100 samples of interarrival time values.

The partition of $\mathbb{R}^+$ used to classify the k previous values of $IAT[n]$ and estimate the conditional distribution according to this classification will have $h = 10$ intervals. Edges will be logarithmically spaced between $t_{min} = 10^{-6}$ and $t_{max} = 0.1$ seconds.

The effect of using different values of k is analyzed comparing the upper bound for the interarrival time of each packet estimated form the distribution of $IAT[n]$ conditioned to the previous k values, with an upper bound estimated from a non conditional distribution, which will based on the same number of samples on which the conditional distribution was based. Since this number of samples is reduced while k increases (see discussion in section 3.4.3), the value of the upper bound estimated from a non-conditional distribution needs to be recalculated for every value of k.

For every subset of traces to/from the same UE, containing at least 5000 packets, we evaluate the percentage of packets whose interarrival time is below their estimated upper bound. Two upper bounds are estimated, so there will be two percentages per subset of traces. Figure 39 shows the average values of both percentages.

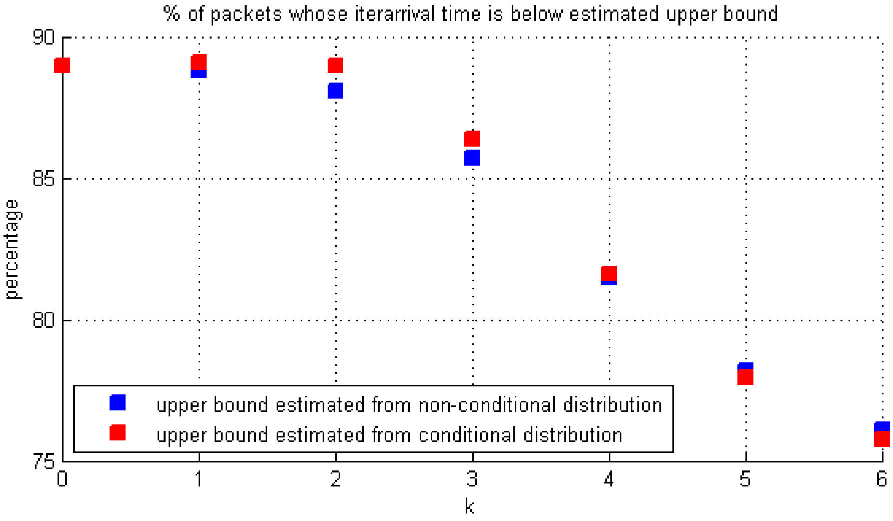


Figure 39: Percentage of packets whose interarrival time is below its estimated upper bound, for different values of k

Figure 39 shows that the evolution of both percentages is similar for different values of k. The percentages are below the ideal value of 90% due to the fact that the number of samples considered to estimate the distributions is finite. This number decreases as k increases, this is why the distance between the percentages and the ideal value of 90% increases while k increases.

For every subset of traces, we evaluate the difference between the average values of both sets of upper bounds. For every subtrace there is one value. A negative value of this parameter shows that the average value of the set of upper bounds estimated from conditional distributions is higher than the average value estimated from non-conditional distributions. Since the average value of a set of upper bounds is supposed to be as low as possible, a negative value of this parameter would mean that upper bounds estimated from non-conditional distributions would be better than the ones estimated from conditional distributions. Figure 40 shows how this difference between average values is distributed among the subsets of traces to/from the same UE.

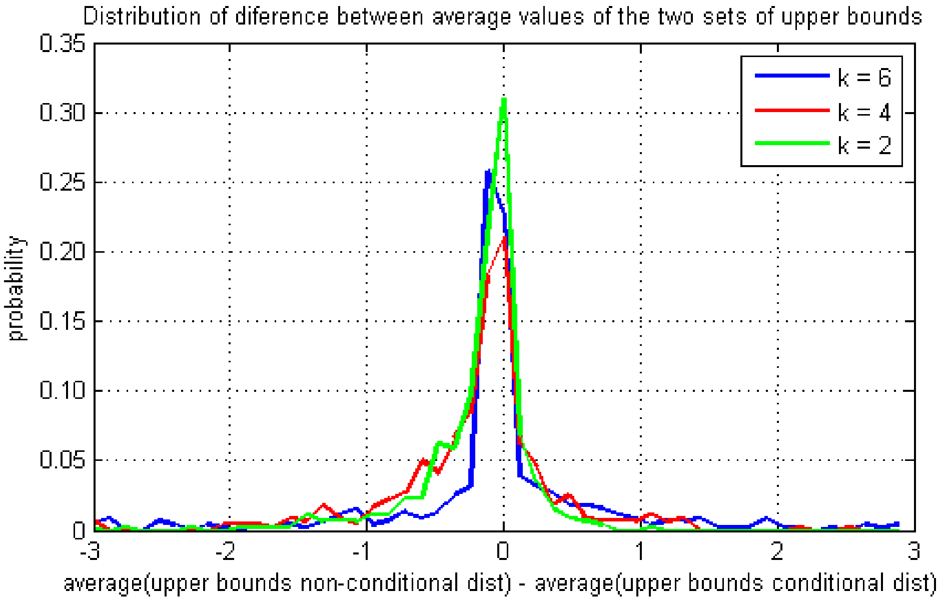


Figure 40: Distribution of the difference between average values of both sets of upper bounds

Figure 40 shows the presence of subsets of traces in which the value of the difference between average values is negative. This means that in those sets of traces, for most of the packets, the upper bound estimated from the non-conditional distributions is below the value estimated from the conditional distribution. On the positive side, the blue line, corresponding to the highest analyzed value of k, is above the other lines for most of the time. This means that with a higher k, a given value of positive difference can be achieved in more subsets of traces. Also, the higher the value of k, the greater the positive difference. Figure 41 shows the average value of both sets of upper bounds estimated for the subset of traces with the highest observed difference between average values.

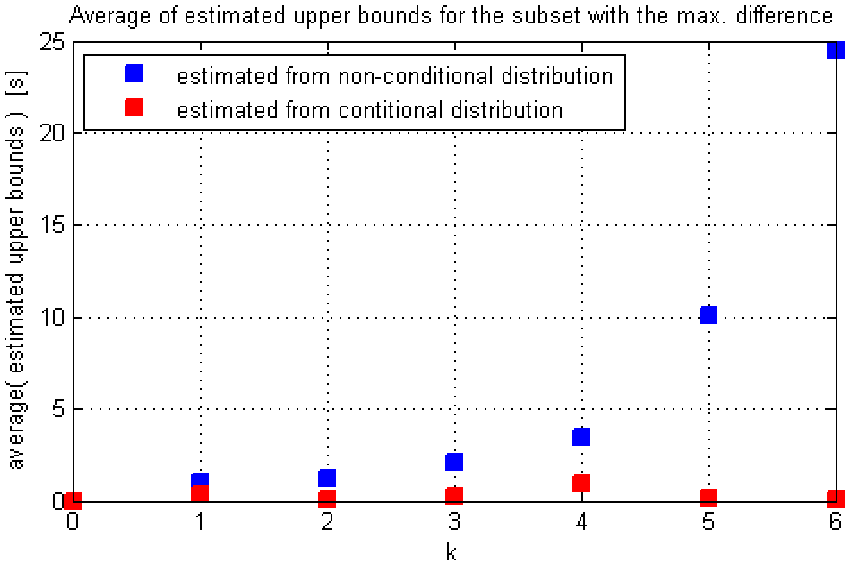


Figure 41: Average value of the two sets of upper bounds for the subset of traces with the highest observed difference between averages

We have a given number of previous values of $IAT[n]$ from which an upper bound for $IAT[n+1]$ is estimated. Using non-conditional distributions, these previous values would be $IAT[n], IAT[n-1], IAT[n-2], \dots$, and they would give the results shown in the blue line. Using distributions of $IAT[n]$ conditioned to the previous k values, the values from which the estimation is made are interarrival times of previous packets that arrived when the interarrival times of their previous k packets were in the same ranges that $IAT[n], IAT[n-1], \dots, IAT[n-k+1]$. They give the results shown in the red line. For certain sets of traces, the higher k is, the better the upper bound can be estimated, compared to the upper bound that would be obtained from the same number of samples, but with samples not meeting the conditions of having their previous k values on the same ranges.

4.4.3. Different partitions of $\mathbb{R}^+$

The partition of $\mathbb{R}^+$ allows the classification of the values of $IAT[n]$. ‘Classify’ is to determine which interval of the partition the value belongs to. According to the classification of its previous k values, the value of $IAT[n]$ will be stored in the appropriate vector, and, after that, according to the classification of $IAT[n], IAT[n-1], \dots, IAT[n-k+1]$, a vector is selected, and an upper bound for $IAT[n+1]$ is estimated from the samples stored in it.

The effect of using different values of h in both linear and logarithmic partition is analyzed. For this analysis we use the following values for the different parameters (see section 3.4.2 for details):

$k = 3$. The condition of the samples considered to estimate the upper bound will involve their previous three interarrival times.

$p = 0.9$. This means that the estimated upper bound of the interarrival time of the next packet for every packet will be the quantile of the distribution of $IAT[n]$ conditioned to the previous $k = 3$ samples for a probability of 0.9. Ideally, 90% of the packets will have an interarrival time which will be below their estimated upper bound.

$m = 130$. Conditional distributions will be estimated taking, at most, 130 samples of interarrival time values.

For every subset of traces, containing at least 5000 packets, we evaluate the percentage of packets whose interarrival time is below their estimated upper bound, and the average value of all the estimated upper bounds.

Figure 42 and Figure 43 show that, for values of the average value of estimated upper bounds, for different values of h equal or greater than 10, linear partitions give better results than logarithmic as more packets arrive within the estimated time, and this estimated time is lower. However, the best observed result corresponds to a logarithmic partition, with $h = 5$.

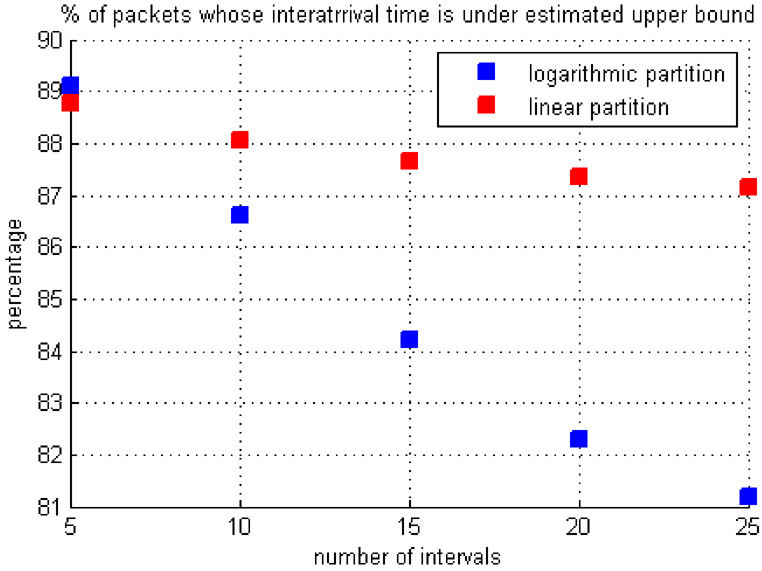


Figure 42: Percentage of packets whose interarrival time is below its estimated upper bound, for different partitions

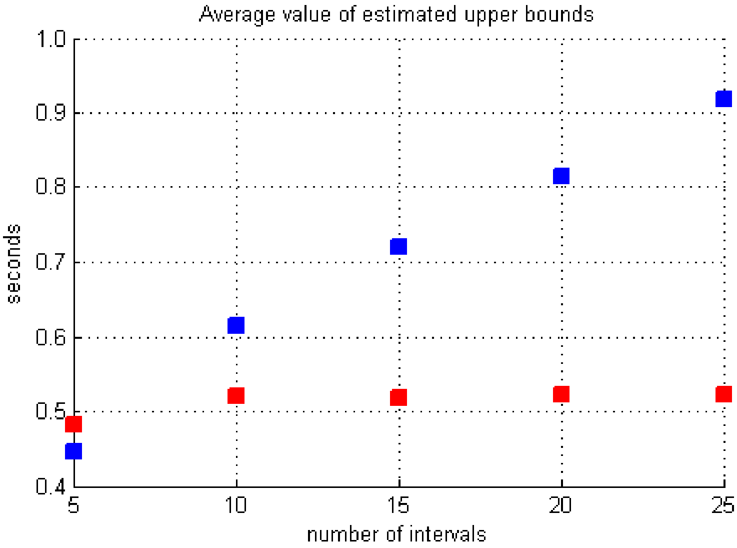


Figure 43: Average value of estimated upper bounds, for different partitions.

Figure 42 and Figure 43 show that, for values of Average value of estimated upper bounds, for different values of h equal or greater than 10, linear partitions give better results than logarithmic: higher number of packets arrive within the estimated time, and this estimated time is lower. However, the best observed result corresponds to a logarithmic partition, with $h = 5$.

5. Conclusions and Future work

In this chapter we present the conclusions extracted from our work, along with suggestions about further work to be done in this area.

5.1. Conclusions

We present the conclusions of this work in terms of answers to the ‘high-level research questions presented in section 1.1.3.

How can traffic patterns be characterized and recognized?

We have proposed a way of detecting sequences of events in the traffic of the UEs which tend to be repeated in time. The estimation of the probability p_A proposed in section 3.3.4 is based on previous events in the traffic of a UE. The table used for the estimation keeps the count of occurrences of each possible sequence of events. A pattern is a sequence of events is repeated in the traffic of this UE many times over time. In section 4.2 we present that certain sequences of events have tendency to be repeated in time. We show that this tendency is not uniform for all the UEs. The tendency of some sequences to be repeated is stronger in some UEs, while in others it is weaker (see Figure 21 for details).

What information can be extracted from the presence of patterns in the traffic?

In section 3.3.3 we reformulated this question in terms of events and the probabilities p_A and p_{IA}, as ‘Can we extract from the presence of patterns information in order to calculate values for p_A and p_{IA} that maximize the number of correct predictions?’.

If it is known that a certain sequence of events is repeated over the time, and this sequence is identified, then it may be possible to predict which is going to be the next event in a near future. In section 3.3.4 we describe a way of estimating the probability of the next event in the traffic in a UE, activity or inactivity. We identify a sequence of events with the event buffer, and in the table we have the number of times each the possible events followed the sequence. The value p_A is estimated from this table, and according to this value we predict the next event, activity or inactivity, flipping a weighted coin.

Figure 23 shows the fraction of correct predictions we get for the different processed subsets of traces, for a static application burst threshold. The closer the fraction of correct prediction is to 1, the stronger is the relationship between the sequence of events in the buffer and the next event. Figure 23 shows that there are UEs in which this relationship is very strong, with values of the fraction of correct predictions near 1, while in others the relationship is weaker. We explore the correlation between the tendency of sequences of events to be repeated and the fraction of correct predictions (Figure 44).

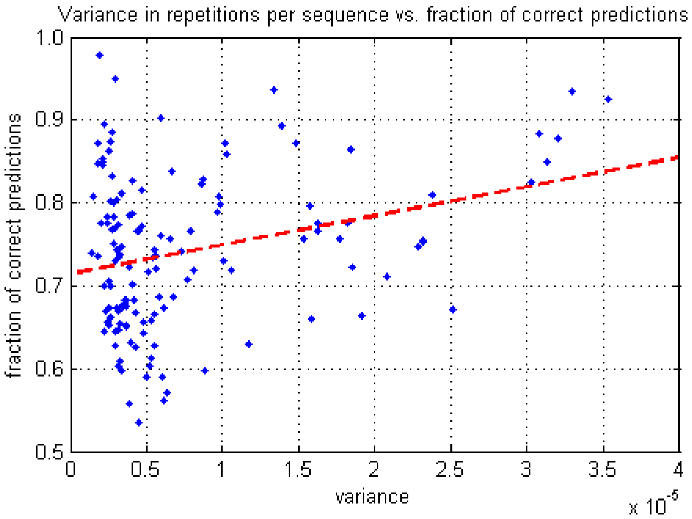


Figure 44: Correlation between variance in fraction of occurrences of sequences and fraction of correct predictions

We can see in Figure 44 that the fraction of correct predictions tends to be higher when the variance in the fraction of occurrences of sequences in the traffic of the UE is higher. And the variance will be higher if there are sequences in the traffic with tendency to be repeated. Hence, from the patterns it is extracted information which increases the number of correct predictions about the next event in the traffic of a UE.

We can also extract from past events in the traffic an upper bound of the interarrival time of the next packet (see description in 3.4 and analysis in section 4.4).

How can the channel scheduling strategy and the DTX/DRX be adapted to the traffic patterns in practical terms?

Channel scheduling could be adapted in the terms described in section 3.3.2. At every prediction moment, the probability p_A is estimated as described in section 3.3.3, based on the event buffer and the table. p_A is our estimation of the probability that the next event is an activity event. Then, we predict the next event. Our prediction is the result of flipping a weighted coin, with the weight p_A for activity event and $p_A = 1 - p_{IA}$ for inactivity event. If an activity event is predicted, network keeps resources allocated for the UE. The resources will be released after a fixed time (static τ_{AB}) or a dynamically calculated time (static τ_{AB}), in case there is no activity of the UE. During this time, the UE remains in Cell_DCH state. If an inactivity event is predicted, network releases the resources allocated to the UE, without waiting any more, just at the prediction moment. At this moment, the UE transitions to a lower power consumption state. Benefits of this are presented in section 4.3.

Knowing an upper bound of the interarrival time of the next packet can help to decide between keeping the UE in Cell_DCH state or transition to a lower power consumption state. After transmission of a packet an upper bound is estimated. If after the estimated time there are not packet arrivals, network can consider that there will be no activity in a near future, and transition the UE to a lower power consumption state.

How much can we improve the network throughput by knowing traffic patterns?

If we adapt the channel scheduling strategy as explained above, we reduce the time a UE is in Cell_DCH without producing data to send or receive. If a UE is in Cell_DCH state, the network reserves resources for this UE (time slots in the physical layer), which cannot be used by any other UE. These resources of a cell are limited, hence, the number of UEs in Cell_DCH in a cell is limited. RNCs run algorithms to manage these limited resources as they receive requests from UE to set up a dedicated connection. If we reduce the time of inactivity of the UEs in Cell_DCH then we will avoid wasting these limited resources, and their availability will increase. This increase in the availability of the resources of a cell will be proportional to the average value of the reduction of the inactivity time (see Figure 24, Figure 29 and Figure 33). The higher is the reduction in the inactivity time, the lower it will take to transition the UE to Cell_DCH state. This means the time t_{switch}, introduced in section 3.3.6 will be reduced, and the burst throughput will increase. The higher is the size of the buffer, the higher is the inactivity time reduction.

However, in our analysis we consider t_{switch} is constant. This allows us to observe the effect of the wrong predictions in the burst throughput perceived by the UE. If an inactivity event is predicted for a UE, the network resources allocated to this UE are released. If this UE generates activity (uplink or downlink packet burst) after the resources are released, within the period of time τ_{AB}, the packet burst will experience an extra delay of t_{switch} due to the reallocation of resources. Note that, if the burst arrives after τ_{AB}, the packet burst will suffer the delay regardless the previous prediction: even thought the network had predicted activity, a timer would have released the resources after τ_{AB}. Since we do not consider the reduction in t_{switch}, the value of the average burst throughput we get for a UE with a prediction system with an event buffer of size $n \geq 1$ is lower than or equal the value of the burst throughput of this UE in the reference system described in section 4.3.1. This is the reason why we talk about 'burst throughput reduction' for a UE, as a cost we pay for having the prediction system, due to the wrong predictions of inactivity. The larger is the size of the event buffer, the lower is the cost paid in burst throughput reduction for each UE (see Figure 25 and Figure 30). This is a proof of the relation between past events in the traffic of a UE and the next event at a certain moment.

How large are the potential battery savings we can achieve?

Cell_DCH is the RRC state in which the battery consumption of the UE is the highest. Our prediction system reduces the time a UE is in Cell_DCH state. Hence, the prediction system allows this UE a reduction in the battery consumption proportional to the reduction of its inactivity time while in Cell_DCH. This reduction is higher in those UEs which a predictable behavior of their traffic (see Figure 26, Figure 31, and Figure 35). The battery saving achieved in absolute terms will depend on each particular UE (brand, model, type of battery, etc.).

We have shown that with a prediction system with dynamic calculation of τ_{AB} (analyzed in section 4.3.3) it is possible to achieve reductions of the inactivity time of UEs in Cell_DCH state much bigger than the reductions achieved using a static value of τ_{AB} (system analyzed in section 4.3.2). However, the fraction of correct predictions is higher in the system with static τ_{AB}. This suggests that the benefit obtained by using the prediction system with dynamic calculation of τ_{AB} is not due to the correct predictions, but it is mostly due to a configuration of the parameter τ_{AB} which is adequate to each particular UE at every time.

5.2. Future work

Further work about the prediction system should focus in a realistic evaluation of the battery consumption reduction and network throughput. Battery consumption of a UE will be reduced proportionally to the reduction of the inactivity time of the UE while in Cell_DCH. But our prediction system could increase the number of transitions to Cell_DCH. These transitions may generate peaks of battery consumption. The reduction should be enough to compensate these peaks. It also should be evaluated how does affect to the scheduling algorithms in RNCs the reduction of the inactivity time of the UEs in Cell_DCH, specially in terms of reduction of the delay t_{switch}.

There is an issue about the network throughput that we have not considered. When a packet arrives and the UE does not have resources allocated, it will be queued. When the resources are allocated to the UE again, all the queued packets will be delivered to the UE. The prediction system will reduce the time a UE is in Cell_DCH state; hence, the number of queued packets will be higher. Therefore, a higher number of packets will have to be delivered to the UE when it transitions to Cell_DCH. The 'burstiness' of the traffic will be higher. The Hurst parameter will increase. One could think that the network resources will be used more efficiently, because more information will be transferred through the radio link each time the UE gets the resources. Further studies should focus on this.

Reduction in t_{switch} and increase in network efficiency are the positive effects of the prediction system. The negative side is the increase in unnecessary signaling overhead due to wrong inactivity predictions (see section 3.3.2). It can increase the time it takes to deliver packets to the UE. If the positive effects do not compensate the negative effect of the wrong predictions, then the users will experience a reduction in their burst throughput, specially those UEs with an unpredictable traffic behavior. It would be necessary to evaluate this possible reduction in terms of user experience: do the user notice the reduction in burst throughput? If it is not a significant reduction, the benefits got in battery consumption reduction may compensate this drawback; if it is a significant reduction, it will not be worth to implement the prediction system.

If it is decided that it is worth to implement the prediction system, further discussion should focus on what is the best place in the UMTS network architecture to implement prediction system. Since Node B is the network equipment closest to the UE, it could be a good candidate. Queued packets would reach the UE with the minimum possible delay as soon as network resources are reallocated to it. The prediction system could also be implemented in the RNC, the equipment which directly controls the channel scheduling. In any case, it should be analyzed if the equipment can deal with the work load required by the prediction system. Note that RNCs already have to handle power measurement reports for every UE, handovers, retransmissions, etc., and it may be better to implement the prediction system in equipment where the work load is not so high.

If results of the evaluation show that it is not worth to implement this prediction system, then further work should focus on how to improve it. One way to improve it could be to involve length of the bursts in the predictions, as an input to the system. Another way could be to let the system predict *only* if the sequence in the buffer is one of the most repeated sequences. Sequences of events have a bounded length (see Figure 22)Figure 22: Distribution of the length of the most repeated sequence.. It would be possible to enumerate these sequences, examine which particular ones are most common and only take decisions if the current sequence of the UE is one of them.

Predictions on length of application burst may help the network to decide if a UE should transition to Cell_DCH in order to transmit or receive, of if it can communicate in Cell_FACH.

The mechanism of estimation of upper bounds of interarrival time of packets should be improved, in order to take into account only values within a certain range. Note that low values of interarrival time are not interesting, since UEs will not turn off the radio if the probability of getting another packet in a very short time is high - this would increase delay and have little effect on power consumption. Very high interarrival times are also not interesting, since the usual timers would have shutdown the radio before then.

References

[1] Harri Holma and Antti Toskala (editors). *WCDMA for UMTS – HSPA Evolution and LTE*. Fourth Edition, John Wiley & Sons, Ltd., 2007, Chichester, UK, 539 pages, ISBN 9780470319338, DOI: 10.1002/9780470512531.

[2] Erik Dahlman, Stefan Parkvall, Johan Sköld, and Per Beming. 3G evolution: HSPA and LTE for Mobile Broadband, First edition, Academic Press, 2007, 496 pages, ISBN 9780123725332.

[3] Toni Janevski and Kire Jakimoski. *Comparative Analysis of Packet Scheduling Schemes for HSDPA Cellular Networks*. Telfor Journal, ISSN 1821-3251, Telecommunications Society, Belgrade and Academic Mind, Belgrade, Volume 1, Number 1, 2009, http://journal.telfor.rs/Published/No1/No01_P01_fin.pdf

[4] Timothy Neame. *Characterisation and Modelling of Internet Traffic Streams*. Doctoral dissertation, Department of Electrical and Electronic Engineering, University of Melbourne, February 2003, 225 pages, http://ww2.ee.unimelb.edu.au/multimedia/research/cubin_TimNeame_Thesis.pdf

[5] Mark E. Crovella and Azer Bestavros. *Self-Similarity in World Wide Web Traffic. Evidence and Possible Causes.* Computer Science Department, Boston University, 1996.

[6] Ashok Erramilli, R. P. Singh, and Parag Pruthi. *An application of deterministic chaotic maps to model packet traffic*. Bell Communications Research, 1995.

[7] R. G. Clegg, et.al. *Criticisms of modelling packet traffic using long-range dependence (extended version)*. Journal of Computer and System Sciences, 2010.

[8] W.E. Leland, M.S. Taqqu, W. Willinger, and D. V. Wilson. *On the Self-Similar Nature of Ethernet Traffic*. SIGCOMM Computer Communication Review, 1995.

[9] W.E. Leland, M.S. Taqqu, W. Willinger, and D. V. Wilson. *On the Self-Similar Nature of Ethernet Traffic (extended version)*. IEEE/ACM Transactions on Networking, Volume 2, Issue 1, 1994.

[10] Richard G. Clegg. *A Practical Guide to Measuring the Hurst Parameter*. Proceedings of 21st UK Performance Engineering Workshop, School of Computing Science Technical Report Series, University of Newcastle, 2005, http://www.richardclegg.org/pubs/rgcpew05.pdf

[11] Guillaume Collin and Boris Chazalet. *Exploiting cooperative behaviors for VoIP communication nodes in a wireless local area network*. Department of Communication Systems, Royal Institute of Technology (KTH), 2007, http://web.it.kth.se/~maguire/Boris-Chazalet_and_Guillaume-cooperative-behaviors-final-report-20070816.pdf

[12] 3GPP Technical Specification TS 25.211: Physical channels and mapping of transport channels onto physical channels (FDD) (Release 10). http://www.3gpp.org/ftp/Specs/archive/25_series/25.211/

[13] 3GPP Technical Specification TS 25.201: Physical layer - General description (Release 10). http://www.3gpp.org/ftp/Specs/archive/25_series/25.201/

TRITA-ICT-EX-2011:174

www.kth.se