Examensarbete

Mobile services based traffic modeling

Kristoffer Strengbom

LiTH-MAT-EX–2015/03–SE
Mobile services based traffic modeling

Mathematical Statistics, Linköpings Universitet

Kristoffer Strengbom

LiTH-MAT-EX–2015/03–SE

Examensarbete: 30 hp

Level: A

Supervisor: Tomas Lundborg,
DURA Systems & Technology, Ericsson

Examiner: Torkel Erhardsson,
Mathematical Statistics, Linköpings Universitet

Linköping: March 2015
Abstract

Traditionally, communication systems have been dominated by voice applications. Today with the emergence of smartphones, focus has shifted towards packet switched networks. The Internet provides a wide variety of services such as video streaming, web browsing, e-mail etc, and IP traffic models are needed in all stages of product development, from early research to system tests. In this thesis, we propose a multi-level model of IP traffic where the user behavior and the actual IP traffic generated from different services are considered as being two independent random processes. The model is based on observations of IP packet header logs from live networks. In this way models can be updated to reflect the ever changing service and end user equipment usage.

Thus, the work can be divided into two parts. The first part is concerned with modeling the traffic from different services. A subscriber is interested in enjoying the services provided on the Internet and traffic modeling should reflect the characteristics of these services. An underlying assumption is that different services generate their own characteristic pattern of data. The FFT is used to analyze the packet traces. We show that the traces contains strong periodicities and that some services are more or less deterministic. For some services this strong frequency content is due to the characteristics of cellular network and for other it is actually a programmed behavior of the service. The periodicities indicate that there are strong correlations between individual packets or bursts of packets.

The second part is concerned with the user behavior, i.e. how the users access the different services in time. We propose a model based on a Markov renewal process and estimate the model parameters. In order to evaluate the model we compare it to two simpler models. We use model selection, using the model’s ability to predict future observations as selection criterion. We show that the proposed Markov renewal model is the best of the three models in this sense. The model selection framework can be used to evaluate future models.

**Keywords** UMTS networks, Internet Protocol, TCP, UDP, Fast Fourier transform, service behavior, Markov renewal process, kernel estimation methods and model selection.

**URL for electronic version:** http://liu.diva-portal.org/smash/record.jsf?pid=diva2:798424
Acknowledgements

I would like to thank my supervisor Tomas Lundborg at DURA Systems and Technology at Ericsson for providing the idea of the thesis, his guidance and helpful feedback during the work of the thesis. I would also like to thank my supervisor and examiner Torkel Erhardsson at MAI, Linköping University for his feedback and valuable ideas.
Nomenclature

Most of the recurring abbreviations and symbols are described here.

Symbols

\[ f(x) \] Probability density function.
\[ F(x) \] Cumulative distribution function.
\[ K(y) \] Kernel function.
\[ p_{i,j} \] Transition probabilities.
\[ Q_{\text{pred}} \] Theoretical measure of a model’s ability to predict future observations.
\[ \hat{Q}_{\text{pred}} \] Estimate of \( Q_{\text{pred}} \).

Abbreviations

CDF        Cumulative Distribution Function
DFT        Discrete Fourier Transform
FFT        Fast Fourier Transform
HTTP       HyperText Transport Protocol
GSM        Global System for Mobile Communication
IP         Internet Protocol
PPP        Point-to-Point Protocol
RNC        Radio Network Controller
RTP        Real-time Transport Protocol
TCP        Transmission Control Protocol
TTI        Transmission Time Interval
UDP        User Datagram Protocol
UMTS       Universal Mobile Telecommunication Systems
UTRAN      UMTS Terrestrial Radio Access Network
VoIP       Voice over IP

Strengbom, 2015.
## Contents

1 Introduction .............................................. 1
   1.1 Background ........................................ 1
   1.2 Objective .......................................... 2
   1.3 Topics covered .................................... 2

2 Preliminaries ......................................... 3
   2.1 Evolution of Mobile Internet ..................... 3
   2.2 TCP/IP protocols .................................. 5
      2.2.1 Application Layer ............................ 6
      2.2.2 Transport Layer ............................. 6
      2.2.3 Network Layer ............................... 7
      2.2.4 Data Link and Physical Layer ............... 8
   2.3 Log Data ........................................... 9
      2.3.1 User Sessions ................................ 10
      2.3.2 Distribution of Services .................... 10
   2.4 Data Extraction .................................... 10
   2.5 Related Work ...................................... 12

3 The Modeling Approach ............................... 15
   3.1 Deterministic vs Random behavior ................ 16
   3.2 Fourier Analysis .................................. 17
      3.2.1 Discrete Fourier Transform .................. 17
      3.2.2 Fast Fourier Transform ....................... 18
      3.2.3 Fourier transformation of log data .......... 18
   3.3 Range of Frequencies .............................. 19
   3.4 Method ............................................. 20

4 Video Streaming ....................................... 21
   4.1 YouTube - Android Media Player ................. 21
      4.1.1 Separation of behaviors ...................... 22
      4.1.2 Parameters of the Model ..................... 24
   4.2 YouTube - iOS Media Player ..................... 25
   4.3 YouTube - RTP ..................................... 26
      4.3.1 Separation of behaviors ...................... 26
      4.3.2 Parameters of the model ...................... 27
      4.3.3 Importance of finding deterministic behaviors 29

5 Other Services ....................................... 33

Strengbom, 2015. xi
5.1 Software Downloads ............................................. 33
  5.1.1 Parameters of the model .................................. 35
5.2 Web Browsing .................................................. 35
5.3 Problems finding deterministic behaviors ................. 37

6 Mix of Services .................................................... 41
  6.1 Model ............................................................ 42
  6.2 Model Selection ................................................ 45
    6.2.1 Ability to predict future observations ............... 45
    6.2.2 Kernel Estimation Methods ............................. 47
    6.2.3 Alternative Models .................................... 48
    6.2.4 Results ................................................... 48

7 Conclusions ....................................................... 51
Chapter 1

Introduction

1.1 Background

Up until some 15-20 years ago, communication systems were dominated by voice applications. During the 1990s two technologies started to emerge. One of them was the Global System for Mobile Communication (GSM). GSM supported digital transmission of voice data and services such as SMS messaging. The other one was the Internet. The Internet or its predecessors had been around since the 1960s in military research projects, but was now getting available to ordinary people. The Internet allowed for services such as web browsing, e-mail, file transfers, video streaming etc.

Since then the two technologies have become more closely interconnected. We have seen the introduction of both smartphones and 3G and 4G networks allowing for higher bitrates. Thus, Internet traffic is becoming more and more important in cellular networks. The number of users in cellular networks still grows rapidly. Furthermore, the usage per user is growing. This means that models are needed for Internet traffic in cellular networks in order to dimension the networks and test new applications.

For voice communication systems, Erlang models have traditionally been used. These models assume that the inter arrival times of events are independent and exponentially distributed, which leads to a Poisson process model. One of the main advantages is that these models are easy to handle analytically. The reason for the success of Erlang models is that the voice service is limited and well-defined and subject to a low degree of variability. Therefore when there was a need for models for packet switched networks, the same framework was used.

However, in the 1990s it was shown that Internet traffic does not necessarily fit into this framework. Considering the literature on Internet traffic modeling, there are basically three phenomena that the proposed models try to capture: heavy tailed distributions, self-similarity and long-range dependence. We will discuss these later. Furthermore, most of the models consider fixed networks and the aggregated traffic of several users. These models are not necessarily...
applicable to cellular networks due to its different characteristics. Thus, the purpose of this thesis is modeling Internet traffic in cellular networks.

1.2 Objective

The thesis work is defined by two goals:

- Based on IP log data, build a random process generator that creates IP data flow for a number of different services.
- Build a model that randomly combines the generators for each specific service according to a suitable distribution and generates emulated traffic in the same format as the input logs.

A service could be for example watching a clip on YouTube, checking your e-mail, web browsing or using Facebook. Basically, a service is any application that involves Internet traffic. An underlying assumption of the thesis is that different services generate their own characteristic data. Thus, we have a third goal:

- Investigate if this assumption is valid and try out some mathematical methods of finding this service behavior.

Due to the complexity of the problem and time limitations, the original goal of building a random process generator was partly abandoned. The focus has rather been on investigating tools and approaches for building such a generator. Thus, in this report we will not present a finished generator, but instead we present important aspects that future work should take into consideration.

1.3 Topics covered

There are six chapters (and this introduction). Main topics dealt with are:

Chapter 2: We explain the protocols on which the Internet is based and how this could affect the modeling. The log data is also introduced as well as data extraction and multi-level modeling.

Chapter 3: An overview of the proposed model is given. Different application behaviors are discussed. We give a short description of fundamental Fourier analysis.

Chapter 4: Video streaming services and their characteristics are investigated.

Chapter 5: Other types of services are investigated.

Chapter 6: A model for the user behavior is proposed as well as methods for evaluating the model.

Chapter 7: Conclusions.
Chapter 2

Preliminaries

2.1 Evolution of Mobile Internet

In this section we take a look at the ongoing and upcoming evolution of mobile Internet. All the information in this section is taken from [5]. The purpose of this section is to give some background information as well as to show why the modeling approach taken in this report is interesting.

The data in mobile networks have grown tremendously during the last decade and this growth is predicted to continue. During 2013, mobile data traffic grew 81 percent. We find the highest rate of growth in the Middle East, Africa and Latin America. By 2018, the global data traffic is predicted to increase to 11 times the current traffic. This can be seen in Figure 2.1. Furthermore, 526 million mobile devices and connections were added in 2013 and the average smartphone usage grew by more than 50 percent.

Figure 2.1: Evolution of global data traffic.

Strengbom, 2015.
Due to the high bitrates required, video streaming is the dominating application in mobile networks when considering generated traffic. Already today it stands for more than 53 percent of the total global data traffic and this number is expected to grow to more than 69 percent by 2018. Also, other multimedia applications such as audio streaming are expected to grow. In total these two will stand for more than 80 percent of the traffic. This can be seen in Figure 2.2. Obviously it is interesting to analyze and characterize this type of traffic when modeling IP traffic in cellular networks.

![Exabytes per Month](image)

**Figure 2.2:** Evolution of data traffic coming from different types of applications. The figures in parentheses refer to traffic share in 2018.

Another interesting aspect is that not only will the number of users in mobile networks increase, but so will the average amount of data generated by a device. Traditionally, quite a small number of users have generated a large portion of the data. Trends now show that mobile data traffic is being evened out among users. Thus, it is interesting to have a model where you can alter what kind of applications are being used but also how often they are used.

There is not just an increase in the number of smartphones and tablets, but also among machine-to-machine connections and wearable systems. Machine-to-machine connections are for example GPS systems in cars. Wearable devices could be for example fitness trackers, smart watches etc. There will be and has been an explosion of new applications and these applications will probably generate different data. All of this indicates that service based modeling is a reasonable approach.
2.2 TCP/IP protocols

The Internet is a very complex system that connects billions of devices. These devices could be for example PCs, smartphones and servers. In the following, these devices will be called hosts. The Internet is based on a technique called packet switching. Packet switching allows users to share a path in the network at the same time [2]. When information is to be exchanged, the information is divided into smaller packets, also called PDUs (Protocol Data Units). The PDUs contain addressing information and checksums to find transmission errors. Different properties of the PDUs, such as size, quantity and timely correlation, depend on the specific protocol. The PDUs are then transferred between the hosts by routers. This is done independently for the PDUs and they do not actually have to take the same path. Resources only have to be assigned to a single PDU and not to an entire connection. This is in contrast to classic circuit switched networks where resources are dedicated to an entire connection.

In order for the hosts to communicate and exchange information with each other protocols are needed. In this section, we will describe these protocols, which are known as the IP protocols [1]. The protocols are defined on five different layers as shown in Figure 2.3. The different layers are from top to bottom: the application layer, the transport layer, the network layer, the data link layer and finally the physical layer.

Taking a look again at Figure 2.3, we can see how the different layers communicate with each other. We observe that hosts need to have all five layers implemented, while the routers only need the bottom three. The two bottom layers are implemented in the hardware. The transport and network layers are part of the operating system of the device. There is a low degree of freedom in these four layers since different devices must be able to communicate. However, on the application layer the programmer could basically implement his/her own protocols [1].

![Figure 2.3: The five layers in the Internet protocol suite.](image)
2.2.1 Application Layer

The top layer is the application layer. This layer basically describes how information is to be exchanged. The most known example of an application layer protocol is perhaps the HyperText Transfer Protocol, HTTP. This protocol defines how information is to be exchanged between a web browser and a web server. Another example is the Real-time Transport Protocol, RTP. This protocol is used for some multimedia streaming applications [2]. The application layer submits data to the transport layer which handles the transmission. Due to the variability, the application layer will be of particular interest for our models. One should note however that we are more interested in how the application behaves, i.e. finding deterministic behaviors, rather than the application protocol. The application protocol is merely the format that the application uses to exchange information.

2.2.2 Transport Layer

From the application layer data is passed to the transport layer. There are two dominating protocols on this layer and these are TCP, Transmission Control Protocol, and UDP, User Datagram Protocol. The main difference between the two is that TCP allows for reliable communication between two hosts and UDP does not. These two protocols will now be explained further.

Transmission Control Protocol

The reliable protocol is TCP and this is usually called a connection-based protocol. Before any data is sent from the application, a connection is set up between two hosts. This is done by a three way handshake. During this phase the sender basically checks whether the receiver is ready to receive data. When the connection is set up, the hosts are ready to send and receive information. When there is no more data to be sent the connection is terminated using a four way connection tear-down [2]. Let us consider the case of a user web browsing. First, there would be a three way handshake. Then the client would send a request to the server and the server acknowledges this and starts sending the response or data. Finally, the connection is torn down. This procedure is shown in Figure 2.4.

Using timers and acknowledgments the sender can find packet losses and data can be retransmitted. One can note that the TCP guarantees the delivery of all data packets, but it does not make any promises when it comes to delays, order of packets and transmission rate. For further information we refer to [2]. TCP is typically used for applications such as web browsing and e-mail clients where the transmission rate is not prioritized.

The data rate is controlled by two adaptive algorithms in TCP. The flow control informs the sender about the free memory in the receiver’s buffer. In this way the sender knows the maximum amount of data to be sent with the next TCP packet. Furthermore, there is the congestion control. The sender can cut the data rate by half at congestion events. Then the data rate is successively increased.
2.2. TCP/IP protocols

These algorithms will affect both the inter packet arrival times and the size of packets.

User Datagram Protocol

The second most commonly used transport layer protocol is UDP. UDP is a connectionless protocol and is unreliable. Being unreliable means that it provides no guarantee that a packet will actually reach its destination. It is also possible for packets to arrive out of order. The advantage of UDP is that it allows for higher data rates as no control mechanism is used such as acknowledgments and retransmissions. Probably, a large part of the data will reach its destination. This is the reason why UDP traditionally have been used for online gaming, multimedia streaming and VoIP.

2.2.3 Network Layer

The communication between two hosts is handled by the network layer. On this layer, the Internet Protocol (IP) is defined. As we can see in Figure 2.3, this is the top layer on the components between the hosts. Data segments from the transport layer are basically mapped on IP packets. In Figure 2.5 we can see the structure of an IP packet. Most of the information in the data logs (to be described in section 2.3) are taken from the IP packet headers. This information consists of IP addresses, transport protocol, size of packet etc. One could describe this layer as putting a letter in an envelope and writing the address on the envelope. Then the envelope is dropped into the mailbox or network.[2]
We note that the paths taken by individual IP packets between two hosts do not have to be the same. The route taken by an IP packet is calculated using different routing algorithms. Exactly, how this is done is out of scope and of little interest in this thesis, so the reader is referred to [2].

2.2.4 Data Link and Physical Layer

Whereas the network layer is responsible for the routing of packets through the network, the data link and physical layers are responsible for the actual transferring of data from one link to another. Here, we define links as hosts or routers. Examples of data link protocols are Ethernet and PPP. The data link layer is responsible for encapsulating the IP packets into frames and might be able to detect and correct errors that occur in the physical layer. The physical layer on the other hand is responsible of transferring the individual bits between the links. This involves, among other things, modulation methods and what frequencies to use. The protocols on this level depends on the transmission medium. The transmission medium could for example be air or fiber optics. A data link protocol can have many physical layer protocols [2].

The physical layer has some interesting aspects on our modeling since we are dealing with cellular networks. Later in section 3.1, we will talk about finding an application behavior. The available bandwidth in cellular networks will vary which may cause problems detecting the application behavior. The networks considered in this thesis are called Universal Mobile Telecommunication Systems (UMTS) or 3G networks. UMTS facilitate both packet switched services like IP traffic and circuit switched services like traditional telephony. As we can see in Figure 2.6 the system can be divided into three different parts depending on functionality. First, we have the UE or user equipment. This is the device that a person uses to connect to the network and it could be for example a smartphone or a tablet.

Next, we have the UMTS Terrestrial Radio Access Network (UTRAN) that is
2.3 Log Data

Next, we take a look at the IP packet header logs we have at our disposal. The data is recorded between the RNC and the core network. In the logs we can find a great deal of information and here we just mention the most important. Each line in the log represents a packet. First of all, we have the packet arrival time with a resolution of microseconds. Next, we find the transport protocol where $t$ stands for TCP and $u$ stands for UDP. Next, we have the IP addresses and ports of the source and destination. Then we have the packet size in bytes followed by a field indicating if it is a downlink or uplink transmission. This is denoted by a $d$ or a $u$.

![Figure 2.6: UMTS architecture.](image)

Figure 2.7: Example of log data where part of a session is marked in red.

responsible for all radio-related functionality. There are two main elements of the UTRAN. The Node Bs or base stations handles the communication between the UE and UTRAN. They perform channel coding, interleaving, rate adaption etc. There is also the Radio Network Controller (RNC). The RNC handles the management of the radio resources. It controls the base stations in its domain. The RNC is connected to the core network.

The last part is the core network. It is responsible for switching and routing calls and data connections to external networks. These external networks are packet switched networks and circuit switch networks.[8]
The last field gives information on the specific service. Here we can find attributes such as application protocol, functionality, encryption and service provider.

2.3.1 User Sessions

Throughout the rest of this thesis we will use the term user sessions. We define a user session to be all the uplink and downlink data generated by the user of a specific service or application. The separation of log data on user sessions will be based on the quadruple defining a connection. This quadruple consists of

- Source IP address
- Destination IP address
- Source port
- Destination port

Thus, we can first separate the data by users if we also take into consideration the direction of the data. Next, we separate the user data on user sessions. For this purpose we will use the service tag, but exactly how this will be done will differ from service to service. In some cases we use a timeout. If no packets arrive within a certain timeout for a service, we define all packets previously transmitted as a session. The next packet arriving for the same service will be the start of a new session and so on. As an example we have marked a session in Figure 2.7.

2.3.2 Distribution of Services

In Figure 2.8, we can see the distribution of downlink log data on different service categories. There is an asymmetry in the data. A user generates much more downlink data than uplink data. This is why we only consider the downlink. Close to 40% of the traffic is generated by video and audio streaming services. Other downlink heavy types of services are software downloads, web browsing, photo sharing and social networking. It should be noted that quite a large part of the data in the category Other is not assigned to any specific service.

2.4 Data Extraction

Before going into the proposed modeling approach, we take a look at what kind of data that can be measured and modeled. As we will see, this can basically be done on a number of different levels. These levels characterize different aspects of the application and user behavior.

In Figure 2.9, we can see the different levels of modeling. The first level is what we call the application or session level. This level characterizes first of all the user behavior. Here, we can capture how the services are accessed in time.
We also find the size of session data. Furthermore, we can also capture the correlation between different services, i.e. what services are used at the same time.

Next, we have the connection level, at least for services that use TCP. As men-
Chapter 2. Preliminaries

TCP is a connection oriented protocol and during the usage of a service one or multiple connections will be established. The connection data can easily be determined since we know the quadruple. Furthermore, the connection starts with a TCP handshake and ends with a four way tear-down. Thus, we can describe the amount of uplink and downlink data during the connection and we can also measure the arrival rate of new connections.

One of the main characteristics of IP traffic is the burstiness. Thus, we could also consider modeling the bursts. The bursts are defined as packets arriving in clusters and a timeout to separate the bursts [1].

The lowest level consists of individual packet sizes and inter packet arrival times. A straightforward approach to modeling IP traffic, based on the format of the data logs, would be to model the inter packet arrival times and packet sizes. However, this method has its drawbacks. TCP, for example, has its flow control and congestion control algorithms as mentioned earlier. These are algorithms that depend on the path between the two hosts, and thus it will affect individual inter arrival times. Furthermore, in cellular networks there are a lot of variations in available bandwidth which obviously would affect the inter packet arrival times. Thus, the individual inter packet arrival times do not characterize a service in most cases [4]. A problem is that we still want the output of the model to be in the same format as the input, i.e. consisting of packets and time stamps. The best way would be to simulate the behavior of TCP on top of our model. Since we do not have access to this kind of equipment, we still have to generate the inter arrival packet times using the log data. However, if possible we use other parameters on upper levels to model the service.

A model that takes several of these levels into account is what we can call a multi-level model [1]. In the next chapter, we will discuss our proposed model which is a multi-level model. The levels used in modeling a service will differ from service to service. By using a multi-level model we try to capture both the user behavior and the application behavior. Hopefully this will lead to a model that can generate more user realistic data.

2.5 Related Work

At this stage it is interesting to mention approaches taken in IP traffic modeling in the literature. As mentioned in the introduction these approaches often consider phenomena such as self-similarity and long range dependence, LRD. Thus, in this section we will shortly mention these concepts and how our work differs.

Self-similarity basically means a process where the process properties are preserved on different time scales. Mathematically, this is defined by

\[ X(at) = a^H X(t), \ a > 0 \] (2.1)

where \( X(t) \) is a stochastic process, \( a \) is a scaling parameter and the equality is in the sense of distributions. \( H \) is called the Hurst parameter. A typical example of a self-similar process is geometric Brownian motion.
LRD is a related concept that means that the autocorrelation function of the process has a slower decay than exponential. This means that the dependence between two samples of the process decays slower than for a Poisson process. This is one of the reasons why Poisson processes are not used when it comes to IP traffic modeling. Models considering these two concepts are often based on aggregated traffic streams in a fixed network. Our work concerns individual users and therefore a shorter time span. This means that the above concepts are not really suitable for our purpose and we need to find other approaches.

Another used modeling approach is what we can call source models. They consider the underlying source of the traffic. Often you can find models of web browsing where the focus is on parameters such as the number of embedded objects in a web page, the size of these, the time between web page accesses etc. This is closer to what we would like to achieve since it is on a user basis. At the same time these models do not consider that the traffic generated by different services could be more or less deterministic. This will be discussed in the next chapter. Investigating these deterministic services has been a main part of the thesis work and is thus something new that we bring to the table.
Chapter 3

The Modeling Approach

IP packets do not appear out of the blue, they are the result of the data required for specific services. Furthermore, the services are initiated by a user. If we assume that the users’ need to use a specific service and the actual traffic generated by the service are independent stochastic processes, we can propose a model such as the one in Figure 3.1. Clearly, this model is a multi-level model. If we compare this to the discussion in section 2.4, the user level corresponds to the session level. Thus, this level captures the user behavior more or less. The user has the freedom to choose from a wide variety of services and these are described by the service and type levels. The log data can be divided on user sessions for a number of different services. However, it could be the case that even within the same service there could be different types of application behaviors. Therefore, the model differs between different types of behaviors. The service and type level corresponds to the connection, burst and packet level. The parameters to use for the model will differ from service to service.

![Figure 3.1](image.png)

Figure 3.1: The proposed traffic model.
3.1 Deterministic vs Random behavior

One of the main concerns of this master thesis is how to find the services that have a deterministic behavior and separate these from those that are of a more random nature. Actually, this is the main reason for using the proposed modeling approach, that the generated traffic has a deterministic part and a random part. An underlying assumption of the thesis is that different types of services behave in different ways and therefore generate traffic with different characteristics. As mentioned before we are not that interested in application layer protocols but rather the programmed behavior of the application. Since there is a large variety of applications there could be a large variety of behaviors. The behaviors are basically defined by the programmer of the application. Now, the question is: how do we find these deterministic behaviors? Another question one might ask is if this actually is the case? Could it just boil down to different applications having different distributions for file sizes and then downloading the files as fast as possible?

By just looking at typical user sessions for a couple of services, we can identify three different phases of the data transmission. The first one is sending data at maximum available bandwidth. In this case we could consider for example a software download where the user needs the whole file. Another example could be downloading a web page with all its embedded objects. The second phase is what we will call a throttling phase. By this we mean that the application or server controls the data rate in some way. We note that when we talk about throttling phases in the rest of the report, we mean throttling that is due to the application. We do not mean throttling that is related to for example TCP. Here, we could consider multimedia streaming applications. In this case it is often not necessary for the user to download the whole file. The third phase is silent periods. Some applications are more of an update character. A typical example would be an e-mail clients making periodic updates. An example of a session with a throttling phase and a session making periodic updates can be

![Figure 3.2: Example of two sessions for two different services. The left one has a throttling phase and the right is making periodic updates.](image)
seen in Figure 3.2. The derivative of the accumulated data basically corresponds to the bitrate. Thus, a higher slope corresponds to a higher bitrate. As we can see in the left plot, the slope changes after a couple of seconds and the bitrate is lowered. This is throttling.

We notice something in particular about the last two phases. If these phases actually exist this would probably lead to some kind of periodicities in the data. Therefore we turn our attention to a standard tool in mathematics and signal processing: Fourier analysis. Since this will be used throughout the thesis, we devote the next section to important definitions and properties of Fourier analysis.

### 3.2 Fourier Analysis

We start by taking a look at a function \( f \in L_2^p(0, z) \), that is a function that has period \( z \) and for which the integral
\[
\int_0^z |f(t)|^2 dt
\]
exists and is finite. The Fourier series expansion of \( f \) is given by
\[
f(t) = \sum_{n=-\infty}^{\infty} c_n e^{2i\pi n \frac{t}{z}}
\]
where the Fourier coefficients \( c_n \) are given by
\[
c_n = \frac{1}{z} \int_0^z f(t) e^{-2i\pi n \frac{t}{z}} dt
\]
Now, we consider a function defined on a bounded interval \((a, b)\). If we repeat this function periodically and call the resulting function \( \tilde{f} \), then \( \tilde{f} \) will also have a Fourier series expansion. This is called the Fourier series expansion of \( f \) on \((a, b)\).

Considering the above definition of the Fourier series expansion, we define the spectrum of \( f \) as the set of pairs \((\frac{2\pi n}{z}, c_n)\), \( n \in \mathbb{Z} \). Thus, the Fourier series expansion is a decomposition of a function into its frequency components. This means that we can find the frequency content of a finite signal by calculating the Fourier series expansion. But right now we are dealing with continuous functions. In practice, we always have a sampled function and this is why we turn to the discrete Fourier transform.

#### 3.2.1 Discrete Fourier Transform

For the discrete Fourier transform, DFT, we make the assumption that we know the period of the function \( f \) and that we have \( N \) of its values that are regularly spaced over the interval. Thus, we know the following
\[ f\left(k \frac{z}{N}\right) = y_k, \ k = 0, 1, 2, ..., N - 1 \] (3.4)

This means that we have a sampled signal where the samples are evenly spaced by \( \frac{z}{N} \) time units. The DFT is an approximation of the Fourier coefficients of the function \( f \). The DFT of order \( N \) is a transformation

\[ (y_k) \rightarrow (Y_n) \] (3.5)

where \( Y_n \) is given by

\[ Y_n = \sum_{k=0}^{N-1} y_k e^{-2i\pi nk \frac{1}{N}}, \ n = 0, ..., N - 1 \] (3.6)

The DFT is bijective and linear. Furthermore, both the sequences \((y_k)\) and \((Y_n)\) can be complex. In our case we will work with a sequence \((y_k)\) of real samples.

### 3.2.2 Fast Fourier Transform

There is a problem, however, of using the definition of the DFT in practice. In order to calculate \((Y_n)\) we need \((N-1)^2\) complex multiplications and \(N(N-1)\) complex additions. If we have \(10^6\) samples, it would lead to more than \(10^{12}\) additions and multiplications, which is a lot. Luckily, there is an algorithm known as the fast Fourier transform, FFT. The FFT was developed by J. W. Cooley and J. W. Tukey during the 1960’s. By using this algorithm, the cost of the DFT is just of order \(N \log N\). Thus, the FFT allows for efficient computing of the frequency content of a signal [3].

### 3.2.3 Fourier transformation of log data

Our goal is thus to find the frequency content of different user sessions for a certain service. We have a “signal” consisting of timestamps and packet sizes. We are working in discrete time and we could consider a session as a sampled signal. However, we need to preprocess the sessions before taking the FFT. First of all, the FFT requires that samples are evenly spaced. This is not fulfilled in the data logs. We have a time resolution in microseconds, but a packet could arrive at an arbitrary time which means that some samples could be one second apart and some could be three microseconds apart. A simple approach to solving this problem is to introduce zeros between the samples. This procedure is shown in Figure 3.3.

The problem here however is that the zero padded sessions will contain a lot of data points. Consider for example a session of 1000 seconds. Adding a zero for every microsecond would result in a billion samples. Thus, the complexity of the calculations will be extremely high, so something has to be done to reduce the number of data points.
3.3. Range of Frequencies

A simple solution to this problem is to bin the samples. By binning we mean that we divide the time into intervals and the value in each bin will be the sum of the packets arriving during this time interval. By trial, we divided each second into 20000 intervals. Thus, this corresponds to a sampling frequency of 20000 Hz. The procedure can be seen in Figure 3.4. Now, we are finally able to analyze the frequency content of the sessions.

3.3 Range of Frequencies

It is worth noting the interesting range of frequencies. Mobile networks use something called transmission time intervals, TTI. In each TTI a certain amount of data is passed from the upper layers to the physical layer. Possible durations of the TTI are for example 1 ms, 2 ms, 10 ms and 20 ms. The duration of the TTI depends on the device used. One could say that the TTI tells us how often the bitrate can be changed. Since the shortest TTI is 1 ms, we are
not that interested in frequencies higher than 1000 Hz. Inter packet arrival
times shorter than this would then correspond to data sent during the same
TTI. Therefore these shorter inter arrival times are not interesting. Scheduling
algorithms determine which users can transmit data during a TTI and also how
much data that can be sent [7].

3.4 Method

We end this section by summarizing and describing the method to find deter-
ministic service behaviors. First, the data is separated on users and services.
The next step is to, for a specific service, taking the FFT. Sessions having differ-
ent frequency content will be regarded as different behaviors and will therefore
be modeled individually. We take a look at what patterns are causing these
strong frequencies and we try to incorporate this into the model. The presence
of a dominating frequency indicates that there is a strong correlation between
the arrival times of packets. However, there could of course be services with no
dominating frequencies and here we would have to look for other parameters to
model.
Chapter 4

Video Streaming

As mentioned in sections 2.1 and 2.3.2, video streaming is one of the most popular applications in mobile networks and it is predicted to continue growing in the near future. Thus, it is important to understand its characteristics and behavior. We note that out of the ten services generating the most downlink traffic, five are different video services. In this section we will look at a couple of different examples of video streaming strategies and try to model these.

4.1 YouTube - Android Media Player

One of the more popular video applications in mobile networks is YouTube. In this section we investigate the characteristics of YouTube data when using Android Media Player. In the log data this service is generating the most downlink data out of all services. We take a look at its main characteristics and use these to create a model for the user sessions. This service uses HTTP as application protocol and TCP as transport protocol.

Figure 4.1: Comparison of two YouTube sessions using Android Media Player.
First, we take a look at two different user sessions in Figure 4.1. We have plotted the accumulated downlink data against time. This means that the derivative of the plots corresponds to the bitrate. Even though they have the same service tag, the two sessions do not look the same. Both plots have a repeating pattern, the services are alternating between periods of data transmission and silent periods. However, there is a difference. Considering the left plot and the first period of data transmission, we see that there is a steep slope corresponding to a higher average bitrate and then the slope decreases corresponding to a lower average bitrate. For the right plot, the bitrate seems to be constant during all periods of data transmission. This is an example of the different types or behaviors that we are interested in for our model. Most of the sessions of this service correspond to one of these two behaviors. Thus, we would like to separate them and as discussed in section 3.4 we take a look at the frequency content.

### 4.1.1 Separation of behaviors

In Figure 4.2 we have plotted the spectrum of the left plot in Figure 4.1. We observe two things. First of all, there are strong lower frequencies corresponding to the alternating pattern mentioned before. Second, there is a strong frequency content between around 1 Hz and 200 Hz. What we are interested in is the frequency 0.8841 Hz. The other strong frequencies in the spectrum are integer multiples of this frequency. These "harmonics" show up due to the fact that we had unevenly spaced data and introduced zeros between the samples. To find out what is causing the frequency of 0.8841 Hz, we zoom into Figure 4.1 and the result can be seen in Figure 4.3.

![Figure 4.2: Spectrum of a YouTube session using Android Media Player.](image)

Here, we can see an interesting result. Bursts of approximately 67 kB are sent
with a frequency of

\[ \frac{1}{45.45 - 44.32} = 0.8850 \text{ Hz} \]  (4.1)

What’s more, this burst arrival frequency differs from session to session. This has reasonably to do with the video encoding rate. A higher video encoding rate necessarily means that more data has to be transmitted, i.e. if we have bursts of a certain fixed size, a higher frequency would correspond to a higher average bitrate. This is what we call a throttling phase where the application or server seems to be controlling the data rate. In Figure 4.4 we can see the cumulative distribution function, CDF, of burst frequencies for YouTube using Android Media Player. Around 35% of the videos behaving in this way have a frequency of about the same as in our example above. This would then correspond to the lowest possible video encoding rate.

![Figure 4.3: Zoomed part of a YouTube session using Android Media Player.](image)

However, if we take a look at the spectrum of the right plot in Figure 4.1, we get a completely different result. This is shown in Figure 4.5. First, we see lower frequencies corresponding to the repeating pattern. The dominating frequencies are now at 100 Hz. These frequencies probably corresponds to the TTI of the device being 10 ms, and they have nothing to do with the actual service behavior. The spectrum of other services also contain strong frequencies at 100 Hz. This will be further discussed in section 5.1.

Zooming into this session we cannot see any clear bursts like for the other session, but something is still causing these frequencies. For now we can consider the two cases as different behaviors, but we will discuss this further in chapter 5. As has been shown, we can use the frequency content to separate the sessions into different types or behaviors. We just have to pick the dominating frequency. However, the higher frequency content (arguably overtones) are a slight problem. The fundamental frequency is not necessarily the strongest one. Next, we discuss the parameters for the model.
4.1.2 Parameters of the Model

The two different behaviors call for two different models, even though we have the same service tag. Here, we will focus on the behavior containing periodic bursts. During the data transmission period, we have three main characteristics. First, we have the period of higher bitrate and we call this an initial burst. Second, we have the throttling phase with its burst frequency. Third, there is
the size of the bursts which seem to vary a little within each session. On top of this, there is also the alternating pattern. There is first an initial burst, then a throttling phase and then a silent phase. Then this is repeated a number of times. Each data transmission period is started with an uplink activity, so the time between consecutive uplinks will be a parameter. Also the amount of data between consecutive uplinks is a parameter. The CDFs for the size of the initial bursts and throttling bursts can be seen in Figure 4.6. As we can see, most of the throttling bursts are approximately 67 kB.

![CDFs of bursts during the throttling phase and the initial burst.](image)

Figure 4.6: CDFs of bursts during the throttling phase and the initial burst.

A session of this type has a deterministic behavior. The parameters are basically set when the clip is accessed and there is a low degree of randomness. We can also find a physical interpretation of most of the parameters. The initial burst corresponds to a buffering phase and the reason for this is to keep the clip running even if there is some bandwidth congestion during the throttling phase. In this way the user is not affected by the congestion. The burst frequency is related to the video encoding rate. The reason for the throttling phase is that smartphones have a limited memory and that the user may interrupt a clip. If the user interrupts the clip, there would have been a waste of bandwidth if the entire clip was downloaded. Furthermore, the alternating pattern also has to do with memory capacity [6].

Considering the possible levels of modeling discussed previously, we note that this model basically tries to model the bursts of the data. The actual time between each packet is not that interesting. During both types of bursts the data is sent at maximum available bandwidth. The important characteristic of this model is that bursts are being sent at a constant frequency.

### 4.2 YouTube - iOS Media Player

Since we have discussed YouTube for Android Media Player, it makes sense to discuss it for iOS Media Player as well. This is also one of the most downlinked
heavy services. Again, the application protocol is HTTP and the transport protocol is TCP. The behaviors of this service are similar to the Android case. We have an initial burst and then the throttling phase with periodic burst of the same size as before. One difference is that there are no alternations between silent periods and periods of data transmission. This means that we have a long throttling phase after the initial burst.

The frequencies of the burst are shown in Figure 4.7. Here, we can see a difference compared to the Android case. Judging by the lower data rate, iPhone users tend to watch videos at a lower quality. The behaviors of both YouTube using Android Media Player and iOS Media Player corresponds to the findings in [6]. The estimated frequencies of the periodic bursts seem to differ a little, though.

4.3 YouTube - RTP

To illustrate different streaming strategies, we consider one more example. Again, we have a YouTube video service but this time the underlying application protocol is RTP, and the transport protocol is UDP. This service does not generate nearly as much traffic as the previous two, but due to the difference in protocols it is still interesting to make a comparison. The use of UDP basically means that the server controls the data transmission integrity since no acknowledgments or congestion algorithms on IP level are used.

4.3.1 Separation of behaviors

Again, we start by looking at two different sessions in Figure 4.8.
The data seems to be transmitted at a constant rate. In this case there is no obvious difference between the two sessions. We investigate the frequency content of these two sessions. The spectrum of the left plot is shown in Figure 4.9 and the spectrum of the right plot is shown in Figure 4.10. Again, we get the "harmonics" but we are looking for the "fundamental" frequency as before. We can now see a clear difference between the two. The left session in Figure 4.9 has two dominating frequencies at 12 Hz and 21.53 Hz. On the other hand, the right plot has dominating frequencies at 15 Hz and 5 Hz. Perhaps the two frequencies in these cases could correspond to separate coding of video and audio [13]. Furthermore, there are sessions with other dominating frequencies. Using the dominating frequency we can again separate the sessions into different behaviors. The CDF of the most common behaviors is shown in Figure 4.11. Here a behavior is characterized by the dominating frequency. We see that a dominating frequency of 12 Hz is the most common behavior, but there are a number of other behaviors too.

4.3.2 Parameters of the model

For this service the different dominating frequencies correspond to different streaming strategies. In this section, we take a look at the type with dominating frequency 12 Hz. The other types are of course modeled as well, but to limit the discussion we will only consider this type. For all sessions of this service the frequencies do not correspond to periodic bursts, but to periodic packets. Thus, there is a strong correlation between the inter arrival times for individual packets.

We take a close look at this type. What can be seen when looking at the inter packet arrival times is that there seems to be two superpositioned processes. One is the source of 12 Hz and the other one is the source of 21.53 Hz. For
this type we can easily find the source of these two frequencies. In the data there are larger packets sent with an inter arrival time of 0.0833 seconds and smaller packets sent with an inter arrival time of 0.045 seconds. The trace of packets can be divided into two traces based on this. Taking the FFT of both the traces we get the result in Figure 4.12. Some of the frequencies in Figure 4.9 appears in the left plot and some in the right plot. Thus, there actually are two superpositioned processes. All the types of this service seem to behave in this way, but the frequencies and the number of superpositioned processes differ.

The big difference here compared to the previously treated YouTube services is that the dominating frequencies actually correspond to the time between two packets. For those services the periodic bursts were the main characteristics.

Figure 4.9: The spectrum of a YouTube RTP session. The dominating frequencies are 12 Hz and 21.53 Hz.

Figure 4.10: The spectrum of a YouTube RTP session. The dominating frequency is 15 Hz.
4.3. Importance of finding deterministic behaviors

The reason for the proposed modeling approach is, as mentioned earlier, to simulate realistic traffic. As we have seen, there are different behaviors among these video streaming services. Not separating the sessions into different types would generate some kind of session that would never appear in reality. It would be a mixture of all the types. This is not what we would like to achieve.

Furthermore, it is not enough to separate between different behaviors and then generate sessions as if individual packets are independent from one another. The spectra of the sessions show that there are strong correlations in the packet
arrival times. An independence assumption would erase this. As we have seen these sessions often have a more or less deterministic behavior and there could be different phases, such as an initial burst and a throttling phase. This has to be considered when modeling. As has been seen, the FFT is an reasonably adequate tool to find these strong correlations.

We illustrate this by simulating two YouTube RTP sessions, where the dominating frequency is 12 Hz. For the first simulation we have just used the inter packet arrival time and the packet sizes. For the second simulation we notice the fact that it seems to be two superpositioned processes and that we can divide a session into two packet traces corresponding to 12 Hz and 21.53 Hz. First, we generate packets at a constant frequency of 12 Hz. To this we then add a deviation from this frequency for each packet. We get the distribution of the deviations by taking the distribution of the inter packet arrival times for the 12 Hz traces and then subtracting the inter arrival time corresponding to 12 Hz, which is

\[ \frac{1}{12 \text{ Hz}} = 0.0833 \text{s} \]  \hspace{1cm} (4.2)

Then we do the same thing using the frequency 21.53 Hz. The two generated traces are then superpositioned. Next, we take a look at the inter packet arrival times for the second simulation. These can be seen in Figure 4.13 together with the empirical packet inter arrival times.

![Figure 4.13: Comparison of empirical and simulated packet inter arrival times.](image)

As can be seen they both have a similar distribution. What is more interesting however is the spectra of a real user session and a simulated session. We have already seen examples of spectra of real sessions. Therefore we only show the spectrum of the second simulation in Figure 4.14. Clearly, the simulation has captured the frequency content. We compare this to the first simulation where we just used the distributions of the inter packet arrival times and packet sizes and considered each packet independent from one another. The independence
assumption means that we just sample the two distributions randomly to generate a packet trace. The spectrum can be seen in Figure 4.15. As we can see, this model does not contain as much information as the other simulation.

Figure 4.14: Spectrum of simulated session where the frequency content has been used.

Figure 4.15: Spectrum of simulated session where packets are considered independent.

However, video streaming has certain characteristics that other services do not. Web browsing or software downloads for example are very different from streaming. Here, one could suspect that there actually are no strong frequencies in the session data due to an application behavior. They should be of a more random nature. Thus, different types of services should call for different models. We take a look at this in the next chapter.
Chapter 5

Other Services

This chapter is a continuation of the previous chapter. First of all we can say that behaviors with a strong frequency content were found for services such as video streaming, audio streaming and e-mail. The reason for this for audio streaming are the same as described in section 4.1.2. However, for e-mail clients the periodic behavior is caused by the client making periodic updates. Thus, for e-mail clients we have lower frequencies. For example the client can make updates every 5 or 15 minutes. Due to space limitations, we will not cover this in this report. However, there are a lot of services where we cannot find periodicities and we will cover some of these in this section. We will discuss the modeling of software downloads and web browsing, two very popular applications in cellular networks. We also conclude the chapter by considering problems of finding deterministic behaviors using the FFT.

5.1 Software Downloads

Software downloads are heavy downlink services. The two dominating software download services in the data are iTunes and Google Play. Since these two show similar characteristics we will treat them as one in this section. The application protocol is HTTP and the transport protocol is TCP. One main difference from video streaming is that a user needs the entire file. In the video streaming case, the user could stop watching a clip. Thus, for software downloads there is really no need for a throttling phase. In Figure 5.1 we have plotted two software download sessions.

It should be noted that the average bitrate in the left plot is $3.1 \text{ Mbit/s}$ and in the right plot it is $383 \text{ kbit/s}$. We see no throttling phase and the data seem to be transmitted at the maximum available bandwidth. The difference in bitrate could be because the devices support different bitrates. Next, we take a look at the spectra of the two sessions and these are shown in Figure 5.2. As can be seen the dominating frequencies are located at 100 Hz and its overtones in the left plot. For the right session they are located at every 50 Hz. This is similar to some of the video streaming cases. It is interesting to investigate where these
frequencies come from. We observe that two of the possible TTIs are 10 ms and 20 ms. This would correspond to 100 Hz and 50 Hz. We try grouping the packets into bursts. Consecutive packets arriving within less than 1 ms from each other will belong to the same burst. Now, we look at the inter burst arrival time which is defined as the inter packet arrival time of the first packets of two consecutive bursts. We plot the CDFs of the times in Figure 5.3.

We immediately note something interesting. The CDF has the shape of a staircase where the length of each step is 10 ms in the left plot and 20 ms in the right plot. Bursts of packets are arriving at integer multiples of 10 ms and 20 ms. All the packets having an inter arrival time shorter than the TTI basically correspond to a burst. This behavior corresponds to data being sent at the maximum available bandwidth. Clearly, this is the reason for the frequencies at 100 Hz and 50 Hz. However, we should also note that for other sessions the dominating frequencies could be at 500 Hz and its overtones. This corresponds
5.2. Web Browsing

In this section we take a look at web browsing. This service is different from the other services we have studied so far and the main difference is the user activity. The user activity of video streaming for example mainly consist of starting the clip. When web browsing, a user accesses different web pages. Each access results in a download of data. We can see an example of a web browsing session in Figure 5.4.

Each of the steps in the plot is basically triggered by the user. The size of the step will depend on the size of a web page for example. The time between the

Figure 5.3: The CDF of inter burst arrival times for the two software download sessions.

to a TTI of 2 ms. What we basically try to show here is that data are sent at the maximum possible rate. Since the bitrate is almost constant in both cases but differ by a factor 10, we may suspect that the rate is determined by the transmission technology used.

5.1.1 Parameters of the model

Even though the data shows that there are bursts for these services too, this is not because of the application. The time between the bursts and especially the size of a burst will depend on the state of the network and the user equipment. Obviously, the times between the packets inside a burst are not of interest either. This means that we actually just have one parameter that is of interest for these software download sessions and that is the size of the file. Furthermore, there is typically no user generated uplink activities during the download.
Figure 5.4: Typical web browsing session.

steps is what we could call reading or thinking time. Looking at more sessions, we can see this behavior over and over again. We have a similar situation as in the software download case. The user needs to download the whole web page in order to see the entire content. We take a look at the frequency content and this can be seen in Figure 5.5.

Figure 5.5: Spectrum of a web browsing session.

We immediately notice two things about the spectrum. The first one is the strong lower frequencies. These frequencies arguably come from the user accessing new web pages. It is a result of the user and not the application itself. The second thing is the frequencies at 100 Hz. The reason for these frequencies were discussed in the previous section. We can draw the conclusion that web browsing is of a random nature. A user accesses a web page and starts downloading the content. After a certain time a new web page is downloaded and so on. The two main parameters to describe the behavior of web browsing sessions are the web page sizes and reading times.
It should be noted that there are more than 150 different web browsing services in the log data. All of them behave in the same way basically. Based on this, it is reasonable to merge all of these web browsing services into one. Another thing to notice is that other services such as Facebook and Instagram behaves in a similar way. This means that similar models with different distributions of the parameters could be used for all these services.

5.3 Problems finding deterministic behaviors

There are two main problems associated with finding deterministic behaviors using the frequency content of the session. For the first problem, we consider the results for YouTube using Android Media Player and software downloads. Both the right plot in Figure 4.1 and the right plot in Figure 5.1 have a constant bitrate and this rate is approximately 380 kbit/s. Thus, we may suspect that the maximum bandwidth of the user equipments are limited by this value. We investigate this further.

For the behavior in the left plot of Figure 4.1, we saw clear initial bursts and throttling phases. For this behavior the lowest burst frequency during the throttling phase is around 0.85 Hz. Given that a burst is approximately 67 kB, this corresponds to an average bitrate of

$$\frac{67 \times 10^3 \times 8}{1.13} \approx 474 \text{kbit/s}$$

(5.1)

However, a burst is transferred using only a couple of TTIs. The user receives quite large parts of each burst in consecutive TTIs which means that the instantaneous bitrate is significantly higher than the average bitrate. If we consider a user equipment that only supports 380 kbit/s, then it would take more than a second to transfer the whole burst. A burst would be smear out in time since only a small part of the burst can be transferred in each TTI. Thus, the pattern of clear bursts would be "erased". The realized bitrate of 380 kbit/s could still be enough to watch the clip without any interruptions. The frequencies corresponding to the bursts during the throttling phase would then disappear and what is left are the frequencies corresponding to the TTIs. With this in mind, the two behaviors in Figure 4.1 could actually be the same. What we see is not two different service behaviors, but two different user equipment behaviors.

The second problem is also related to the bandwidth. The bandwidth in each cell is limited and shared among the users. Often the users are not alone in a cell. A high load in a cell could again lead to the bursts being smeared in time. The result on the frequency content is that the frequencies of the bursts will become weaker. On the same time frequencies corresponding to the TTIs will become stronger. This leads to problems using the frequency content as a separation criterion. This can actually be seen in Figure 5.6. After approximately 90 s we can see that the service enters a throttling phase. However, the spectrum of this service contains dominating frequencies at 100 Hz.
Chapter 5. Other Services

For YouTube using Android Media Player or iOS Media Player we are still able to find quite a lot of sessions with the characteristics described in sections 4.1.2 and 4.2. A reason for this is that they are popular services and we have several thousands of sessions. Considering the described statistics for the initial burst and burst frequencies in the Android case, we have only used slightly more that 20% of the sessions. In [6] YouTube traffic is measured during night time when there is a low load in a cell. The study is from 2014 and they only report behaviors with initial bursts and throttling phases such as the one in the left plot in Figure 4.1. Considering this and the two problems above one could suspect that there is actually just one behavior for these services. For other less popular services, we do not have access to as much data. This means that even though there is some underlying service behavior we may not be able find it. We would like to have access to more data.

Another problem related to using the FFT to find a service behavior is that for some services the main part of the data is transmitted using the maximum available bandwidth and then a smaller part uses a throttling phase where the bursts are sent with a low frequency. An example of this can be seen in Figure 5.7.

The right plot shows a zoomed part of the session and we can see that there are bursts arriving with a 15 second interval. Taking a look at the spectrum in Figure 5.8, we cannot see a strong frequency content corresponding to 0.0667 Hz. By just looking at the sessions we can, however, see that there is some programmed behavior and therefore use this for the modeling.

The FFT tells us about the frequency content of the signal, but it does not reveal the temporal aspects of the signal. Perhaps a better separation of sessions could be done if we used a tool that provided information in both the frequency domain and the time domain simultaneously. Examples of such tools could be Gabor filters or wavelets. In this report we do not investigate these tools.
5.3. Problems finding deterministic behaviors

Figure 5.7: A video streaming session using Android Media Player. The right plot shows a zoomed part of the session.

Figure 5.8: The spectrum of a video streaming session using Android Media Player.
Chapter 6

Mix of Services

In the previous chapters we have considered models for individual services, such as YouTube or web browsing. However, a smartphone user may use several services at the same time. A user could for example stream audio at the same time as he is web browsing. Thus, we are interested in finding the correlation between services. What services are used at the same time and what services are used after each other? First, we take a look at the user activity during a time period of 1 hour for two users. In Figure 6.1 we have chosen 35 different services and plotted how they are accessed in time. We can also see the duration of data transmission.

Figure 6.1: Accessing of services for two separate user during an hour.

As can be seen User 1 is quite active compared to User 2. We note that each...
row corresponds to a specific service. We start by proposing a model for the mix of services.

### 6.1 Model

Before going into the user behavior model, we first need to make some simplifying assumptions. We will consider a model that takes the arrival of the services into account. The simplifying assumptions are listed below.

- The data generated by the services are independent of how many services are used at the same time. Hence, the mixture of services will just correspond to a superposition of the data generated by the individual services.
- The service to be accessed next depends only on the most recently accessed service. Thus, the next service is independent of what has happened before the most recent access.
- The time between successive accesses does not depend on the time for which the most recent service is being used.

Are these assumptions reasonable? We take a look at the first one. Due to the high variability in bandwidth in cellular networks, the first assumption is reasonable as long as there is enough available bandwidth. Otherwise the services would interfere with each other to a large extent.

For the second assumption, it would probably be better to use more information, i.e. use the last 2-3 accesses or even more. However, if we take a look at Figure 6.2 we see that 35% of the users only use one service. Also, around 60% use less than three services.

![Figure 6.2: CDF of number of used services during an hour.](image)

Thus, the more previous states we take into consideration the more data we will need to estimate the parameters. Considering this it makes sense to use this Markov style approach. In the log data there is an asymmetry between users meaning that a small percentage of the users generate a large portion of
the data. A lot of the users will behave in the same way as User 2 in Figure 6.1.

For the last assumption, it is obvious that the time a service is used will affect the time to the next accessed service. However, taking this into account would lead to a much more complex model. We prioritize simplicity and therefore we do not consider this problem.

Now, we are ready for the model and an overview of our model for the mixture of services is shown in Figure 6.3. Our model contains $m$ different states, where each state corresponds to the last service accessed. Thus, the first column of states is the first service accessed. The second column of states is the second service accessed and so on. Furthermore, we also note that there is an empty or absorbing state. This corresponds to a user not accessing more services during the simulations.

Since the next state only depends on the current state and is independent of time we can define a transition matrix that will be constant. The transition matrix is a stochastic matrix where each row of the matrix sums to 1. It is given by

$$P = \begin{pmatrix}
p_{1,1} & p_{1,2} & \cdots & p_{1,m+1} \\
p_{2,1} & p_{2,2} & \cdots & p_{2,m+1} \\
\vdots & \vdots & \ddots & \vdots \\
p_{m+1,1} & p_{m+1,2} & \cdots & p_{m+1,m+1}
\end{pmatrix}$$
We have that \( p_{m+1,m+1} = 1 \) and all other elements of the last row equals 0.

Also, the time to the next access will depend on the present state as well as on the next state. This means that there is a probability distribution of the inter service arrival time associated with each arch in the figure. We call this model \( M1 \) and this process can be described by the following expression

\[
P(\tau_{n+1} \leq t, X_{n+1} = j | X_n = i, T_n, ..., X_0, T_0) =
\]

\[
= P(\tau_{n+1} \leq t, X_{n+1} = j | X_n = i) =
\]

\[
= p_{i,j} F_{i,j}(t)
\]

(6.1)

where \( X_n \) is the state variable, \( T_n \) is the time of transition, \( \tau_n = T_{n+1} - T_n \) and \( F_{i,j} \) is the CDF of the inter service access times between service \( i \) and \( j \). This is usually called a Markov renewal process. See [12] for further information on Markov renewal processes.

### Estimation of model parameters

We need to estimate the transition probabilities. Each row in the transition matrix sums to one and is basically describing a discrete probability distribution. From the log data we also have a count vector \((n_{i,1}, ..., n_{i,m})\) for each state \( i \). The elements \( n_{i,j} \) of this vector represent the number of jumps from state \( i \) to state \( j \). The total number of jumps from state \( i \) to any other state, including state \( i \), is thus given by

\[
n_i = \sum_{j=1}^{m} n_{i,j}
\]

(6.2)

Given these count vectors we want to estimate the probability distributions of each row. The transition probabilities are estimated by the following expression

\[
p_{i,j} = \frac{\alpha_{i,j} + n_{i,j}}{n_i + 1}
\]

(6.3)

where

\[
\alpha_{i,j} = \frac{1}{n}
\]

(6.4)

We can notice that \( p_{i,j} \) is a Bayesian estimator. Recall that Bayesian inference means that we consider the unknown parameters as random variables, having the prior distribution as their probability distribution. We then find the conditional distribution of the parameters given the observations (the so called posterior distribution). The Bayes estimator is the conditional mean vector of the parameters given the observations. For more information about Bayesian
inference, see [11]. If we use a Dirichlet distribution with all parameters being \( \alpha_{i,j} = \frac{1}{n} \) as the prior distribution, we will end up with a Dirichlet posterior distribution too. Taking the expectation of the posterior distribution we will get the above estimate of the probabilities.

Another estimate of the transition probabilities would be

\[
p_{i,j} = \frac{n_{i,j}}{\sum_{j=1}^{n} n_{i,j}}
\]

with the same notation as before. However, if a jump from state \( i \) to \( j \) does not appear in the data, the probability would be 0. It’s reasonable to believe that all transitions are possible and this is why we use the Bayes estimator instead where all probabilities are nonzero.

Furthermore, we need an estimate of the probability of starting in state \( i \) which we call \( p_{\text{init}}(i) \). This distribution is estimated by considering the number of users starting in state \( i \) divided by the total number of users.

As mentioned earlier there are also probability distributions, \( f_{i,j} \), for the inter service arrival times associated with each transition. These can be estimated using the corresponding empirical distributions, or by kernel estimation methods (see below). For the states where there are no transitions in the data, we simply assume a uniform distribution between 0 and 3000 s. The reason for 3000 s is that it is well above the maximum time between transitions in the data. The CDF of the inter service arrival times is denoted \( F_{i,j} \).

### 6.2 Model Selection

At this stage we have the previously described model for the mixture of services, but we would also like to have some kind of measure to evaluate how good the model is. In this section we will consider model selection which basically means picking the best model in a certain sense, among a collection of alternative models. This is a central problem in mathematical statistics. For the model selection we will need some selection criterion and often used is the model’s ability to predict future observations [10].

#### 6.2.1 Ability to predict future observations

Now, we take a look at the measure used to decide if one model is better than another. First, we discuss the setting. We consider the user activity or mixture of services as outcomes of a stochastic variable \( X \). We have \( N \) number of observations \( x = (x_1, x_2, ..., x_N) \). Each of these observations is considered independent from one another, which means that the behavior of one user does not affect the behavior of another user. Furthermore, we have a collection of models \( M \), which are described a number of parameters denoted by \( \theta = (\theta_1, \theta_2, ..., \theta_p) \). We call \( p \) the size of the model and the size can differ for each model.
The first step is to estimate the parameters of the model and the procedure has been described previously in this chapter for model $M_1$. The estimated parameters are denoted by $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_p)$. The next step is to evaluate the model, but it is not completely clear how one would do this. As mentioned earlier a common approach to selecting a model is to look at the model’s ability to predict future observations. Then we can select the model that seems to be best at predicting. Thus, we want to measure the cost of predicting future observations by using a specific model. In order to do this we choose a loss function. For our loss function we will use the logarithmic likelihood function

$$
Loss(X', \hat{X}) = -\ln L(\hat{\theta}|X') = -\ln f_X(X'|\hat{\theta}) \quad (6.6)
$$

where $X'$ is a future observation and $\hat{X}$ are the predicted future observation under the given estimated model. However, one could also consider other loss functions such as the quadratic loss function

$$
Loss_2(X', \hat{X}) = (X' - \hat{X})^2 \quad (6.7)
$$

These quantities are stochastic and therefore we look at the expected value to get a theoretical value of the cost of predicting. For the ability to predict measure $Q_{pred}$ we get the following expression

$$
Q_{pred} = E(Loss(X', \hat{X})) \quad (6.8)
$$

For our models, it is not straightforward to calculate this quantity. However, we can still estimate it. We assume that our observations are independent and identically distributed. We then divide the work into two stages. We first estimate the model parameters $\hat{\theta}$ and then we estimate $Q_{pred}$ given the estimated parameters. However, we would like to avoid using the same observations for the estimation process and for the evaluation process. There are different techniques for avoiding this such as cross validation and they are especially useful when we have a quite small number of observations. Since we have a large number of observations we use another approach. We divide the observations into two sets. The first set is the training set $x_{training}$ used to estimate the parameters and the second is the test set $x_{test}$ used for the evaluation of the model.

We get the following expression for the estimate $\hat{Q}_{pred}$ of $Q_{pred}$

$$
\hat{Q}_{pred} = -\frac{1}{n} \sum_{i=1}^{n} \ln f_X(x_{test,i}|\hat{\theta}) \quad (6.9)
$$

where $n$ is the number of observations in $x_{test}$. We calculate this value for each model in $M$ and the model having the lowest value will be considered the best model. This means that we are measuring the model’s ability to predict future observations by considering how likely they are. On its own, $\hat{Q}_{pred}$ does not really say much. We have to compare it to the values of the other models too [10].
6.2. Model Selection

6.2.2 Kernel Estimation Methods

Now, we need to have a look at the expression for \( f_X(x_{test,i}|\hat{\theta}) \). We do this by considering an example. Our proposed model \( M1 \) is described by the probability distribution of the initial state, the state transition matrix and the distribution of the time in state \( i \) before moving to state \( j \). We consider an example where we start in state 1 and then we move to state 2 followed by state 3. The times between the transitions are \( t_{1,2} \) and \( t_{2,3} \). We call this observation \( x \). For the likelihood function of this observation we get

\[
f_{X,i}(x|\hat{\theta}) = p_{\text{init}}(1)p_{1,2}f_{1,2}(t_{1,2})p_{2,3}f_{2,3}(t_{2,3})p_{3,\text{end}}
\]  

(6.10)

where \( p_{3,\text{end}} \) is the probability of going from state 3 to the end state. Note that we do not associate a time with this transition.

There is however a problem with the process described above and that is the quantities \( f_{1,2} \) and \( f_{2,3} \). The times between consecutive service accesses should have a continuous distribution. This is a reasonable assumption. On the other hand, we have a finite number of values from the training set \( X_{\text{test}} \). Since we are dealing with continuous distributions the values in the test set will not be the same as the corresponding values in the training set in most cases. Just using the empirical values from the training set, would therefore result in a lot of future observations having zero probability or infinite logarithmic likelihood loss function. Therefore, in order to use the process described above we need to estimate the continuous distributions. Then it would be possible to determine the value of \( f_{X,i}(x|\hat{\theta}) \). Here, we have used kernel estimation methods and these will be described next.

The setting is the following; we have a sequence \( x_1, ..., x_m \) of outcomes of iid random variables. These have the same probability density function \( f(x) \). Now, the problem is to estimate \( f(x) \). The kernel density estimator is given by

\[
f_h(x) = \frac{1}{mh} \sum_{i=1}^{m} K\left(\frac{x - x_i}{h}\right)
\]  

(6.11)

where \( h \), which is a smoothing parameter, is called the bandwidth and \( K \) is called a kernel. The kernel is an even function that integrates to one.

\[
\int_{-\infty}^{\infty} K(y)dy = 1
\]  

(6.12)

Thus, there is a number of different kernels that could be used such as a normal or box kernel. We have used a standard normal kernel

\[
K(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}
\]  

(6.13)

This means that the kernel density estimate at \( x \) basically is the average of \( m \) normal distributions with different means. The means of these normal distributions are the outcomes \( x_1, ..., x_m \). The variance of the normal distributions are
given by $h^2$. The smaller the $h$ is, the more centered the normal distributions are around their means \[9\].

### 6.2.3 Alternative Models

We consider two alternative models which both have a similar structure as the previously proposed model. For the first alternative model called $M_2$, we assume that there is an equal probability to access a service. Furthermore, the time between two accesses do not depend on the two states. Thus, all the times between the access of two consecutive services come from the same distribution. For simplicity we assume that this distribution is uniform between 0 and 3000 s. The process is defined by

$$P(\tau_{n+1} \leq t, X_{n+1} = j | X_n = i, T_n, ..., X_0, T_0) = p_j F_{\text{uniform}}(t) = \frac{1}{K} F_{\text{uniform}}(t) \tag{6.14}$$

where $K$ is the number of services including the end state and the other notation is the same as in the previous section. Thus, for this model we do not estimate any parameters. The structure is the same as the model in the previous section, but there is no transition matrix and the distributions are chosen arbitrarily.

The second alternative model called $M_3$ has exactly the same structure, but the probability of accessing a service is estimated by the empirical service distribution, i.e. how often a service is accessed. The distribution of the time between service accesses can be estimated by the empirical distribution or by kernel estimation methods. This model does actually use the data and thus we might suspect that this model is better. The process is given by

$$P(\tau_{n+1} \leq t, X_{n+1} = j | X_n = i, T_n, ..., X_0, T_0) = p_j F_{\text{empirical}}(t) \tag{6.15}$$

We note again that $F_{\text{empirical}}$ does not take into consideration what services are being accessed.

### 6.2.4 Results

We have approximately 150000 observations. For the training set $x_{\text{training}}$ we used 90 % of the data, i.e. 135000 observations. The rest was used for the test set $x_{\text{test}}$. This was done by just choosing the training samples randomly.

The calculated values of $\hat{Q}_{\text{pred}}$ are given in Table 6.1. As we can see model $M_2$ gives the highest value and is thus considered the worst model. This is what we expected since we have not used any information from the training set. We still included this model for the purpose of comparing the values.

Model $M_3$ is the second best model. Here, we have actually used the training data, but we do not take into consideration that successive accesses of services
could be correlated. For the time between the transitions we do not make a difference between which services are involved.

For model $M1$ or our proposed model we get the lowest value of the ability to predict measure. This is good news and what we expected. This indicates that there probably is a correlation between the accesses of services. Even though it is hard to say how much better the model is in some sense, we can at least say that model $M1$ should be better at predicting future observations.

Each of the models includes a certain amount of information about the training data. We can see that the more information from the training set we include, the better the model is at predicting future observations. But there is of course a possibility that there is a better model than $M1$. Again, it is quite hard to say anything about the value of $Q_{pred}$ itself or the difference between different models. We cannot really say this model is this much better than that model.

$M1$ was actually tested several times using different values for the smoothing parameter $h$ of the kernel estimator. In Table 6.2 we just show the lowest one. We also tried other values such as 25, 50 and 100. The results are shown in Table 6.2. In all cases $Q_{pred}$ is smaller than for models $M2$ and $M3$.
Chapter 7

Conclusions

In this thesis a new approach to IP traffic simulation in cellular networks has been investigated. The work could be divided into two parts. The first part has been investigating the characteristics of different smartphone services. We have shown that IP traffic generated by services such as video streaming, e-mail and audio streaming has a more or less deterministic behavior. The strong frequency content indicates a strong correlation between individual packets or bursts of packets. Depending on the applications where a random process generator will be used, this frequency content could be of great interest. This is especially the case when simulating traffic for applications where the predictability of the traffic is a main concern, such as different machine learning applications. Furthermore, we have created several service traffic generators that takes this frequency content into account.

We have shown that the FFT is an adequate tool for finding these periodic, deterministic behaviors both on a packet and a burst level. However, other deterministic behaviors not having any periodicities can obviously not be found using the FFT. For those cases other methods would have to be used.

The second part of the thesis has been about the behavior of a user. A model has been proposed that describes the way users access services. In addition to this, two simpler models have been proposed for evaluation and comparison. The measure to use for the model selection is the model’s ability to predict future observations. We showed that our model is at least better than the other two in this sense. Even though the proposed models are not that complex, we have proposed a framework for evaluating models. In the future this could be used for testing new models. For our proposed model we have used a Markov renewal process. A straightforward extension of this would be letting the next service access depend on the last two accesses or the last three accesses and so on. The framework could still be used in these cases.

One of the drawbacks of the proposed model is the number of parameters of the model. First of all, we have over 1000 services and if we have a separate model for each of these we would get several thousands of parameters. Second, if we consider the model for the mixture of services, we would have to estimate a million distributions for the time between accesses. The model would not be

Strengbom, 2015. 51
very easy to handle and definitely not analytically tractable. Additionally it would require a large amount of data for the estimation process. It would take a lot of time generating the traffic for a single user and even more to generate traffic for tens of thousands of users. However, several service tags are more or less identical. Therefore some services could probably be merged into one. The estimation of all parameters and distributions would also take a very long time. One would have to consider some trade off between the complexity of the model and the accuracy.
Bibliography


Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years from the date of publication barring exceptional circumstances. The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

Upphovsrätt

Detta dokument hälls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår. Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätt utan en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art. Upphovsmannens ideella rätt innefattar rätt att bli nämnt som upphovsmann i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skyld mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egna art. För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

© 2015, Kristoffer Strengbom