



Undergraduate Project Report 2015/16

[User behaviour analysis based on Deep Packet Inspection]

Name:	[Cheng Qian]
Programme:	[Telecom]
Class:	[2012215104]
QM Student No.	[120721267]
BUPT Student No.	[2012212860]
Project No.	[RN_2860]

Date [2016/05/16]

Table of Contents

Abstract	3
Chapter 1: Introduction	5
Chapter 2: Background	6
2.1 User behaviour analysis.....	6
2.2 Deep Packet Inspection	6
2.2.1 Packet	6
2.2.2 Inspection principle.....	7
2.2.3 Three main recognition technology	8
2.3 Advantage of using DPI on user behaviour analysis.....	9
Chapter 3: Design and Implementation	10
3.1 nDPI platform deployment and testing	10
3.1.1 Open source plans selection	10
3.1.2 Deployment in a real network environment	10
3.2 Real-time grasp the Internet packets and analyses the protocol type.....	11
3.2.1 Execution flow	11
3.2.2 Regular expression summary	16
3.2.3 Chunked packet encoding format in HTTP message and decompression implementation with C	18
3.3 Programming realization of user behaviour analysis system	20
3.3.1 MySQL storage	20
3.3.2 Front-end query system	21
Chapter 4: Results and Discussion.....	26
Chapter 5: Conclusion and Further Work	27
References.....	28
Acknowledgement	29
Risk Assessment	30
Environmental Impact Assessment	31

Abstract

Deep Packet Inspection (DPI) is a software solution that monitors a network's data stream and identifies protocols and applications, inappropriate URLs, instructions attempts and malware by looking deep into data packets. DPI provides important security and translation functions by inspecting incoming packets, reassembling and decompressing them, analysing the code and passing data to appropriate applications and services, which make DPI possible to be a suitable technique for analysing the user behaviour and habits. This project proposed and implemented a whole system from data acquisition to analysis, which contained a nDPI platform on switch, a database and a front-end query system.

深度包检测是一种软件解决方案，被用于监控网络数据流、识别协议及应用，以及非常规 URL、指令和恶意软件。其基于的技术原理是能够深度读取 IP 包载荷的内容。DPI 通过检测、重组和解压缩数据包、分析代码、将数据传递给适当的应用和服务等步骤，提供了极为重要的安全性和转译功能。而这一特性使得 DPI 有可能成为分析用户行为及习惯的有力工具。本项目旨在提出并实现一种从数据采集到分析的完整系统，其中包括了一个部署于交换机的 nDPI 平台、一个数据库以及一个前端查询系统。

Chapter 1: Introduction

In this age filled with tons of information, the data became the most valuable source, especially those data specifically indicate every individual's behaviour and habits. The Internet company pay much more attention to the subject of user behaviour analysis, because it can be applied in several aspects like accurate advertisement operation, recommendation based on user interests and habits. But none of the companies can get the information of all the users in the network by one's self. In the past, Deep Packet Inspection was used by network carrier to monitor integrated network state and block malware packets. However, the characteristic of reading into the load content of packet make DPI a possible way to obtain large amount of user behaviour information. This project is set out to verify this proposal's feasibility. In this project, I finished a nDPI platform on switch, a database and a front-end query system.

This report is organized by several components. Chapter 2 briefly introduces the background of some key technology that this project used, including primary user behaviour analysis and Deep Packet Inspection principle. Chapter 3 describes the design and implementation of nDPI platform, database and front-end analysis system. Chapter 4 shows the result and makes a discussion of the application. In chapter 5, it comes up the conclusion of this report and provides a social opinion towards this technique. The writer also makes a risk assessment and an environmental impact assessment in the last two pages.

Chapter 2: Background

2.1 User behaviour analysis

User behaviour analysis is a common technique used by most of the Internet company to analyse their consumers' tastes and habits. It is usually combined with the use of user click tracking and big data analysis on user logs. Benefited from this technology, those companies who have large amount of user data will be advanced in several directions, like more accurate advertisements and notifications, increased daily active user, etc.

2.2 Deep Packet Inspection

2.2.1 Packet

Packet is the unit of data transmission in TCP/IP protocol. In TCP/IP protocol, data were sent into the stack of protocol and transmitted through each of the network layers and finally serialized as data stream to be sent into the network. TCP/IP protocol works on the network layer and the transport layer of the OSI model. It encapsulates the upper layer of data into TCP/IP data message and then divides into smaller data unit, packet, which will be sent into the network. Figure 2.2.1a shows the encapsulation process.

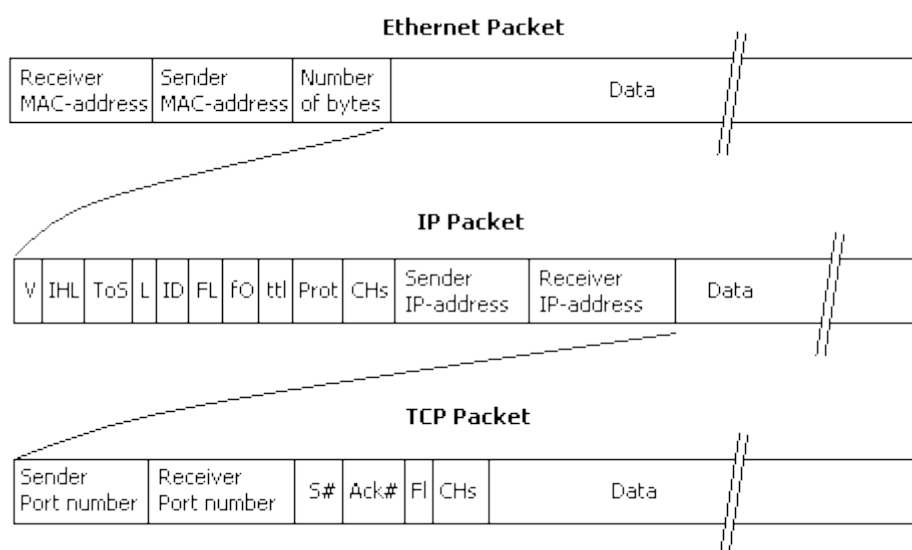


Figure 2.2.1a Packet encapsulation process

[RN_2860] [User behaviour analysis based on Deep Packet Inspection]

The whole network topology can be referred to Figure 2.2.1b. And the routes of packets are shown as the full line.

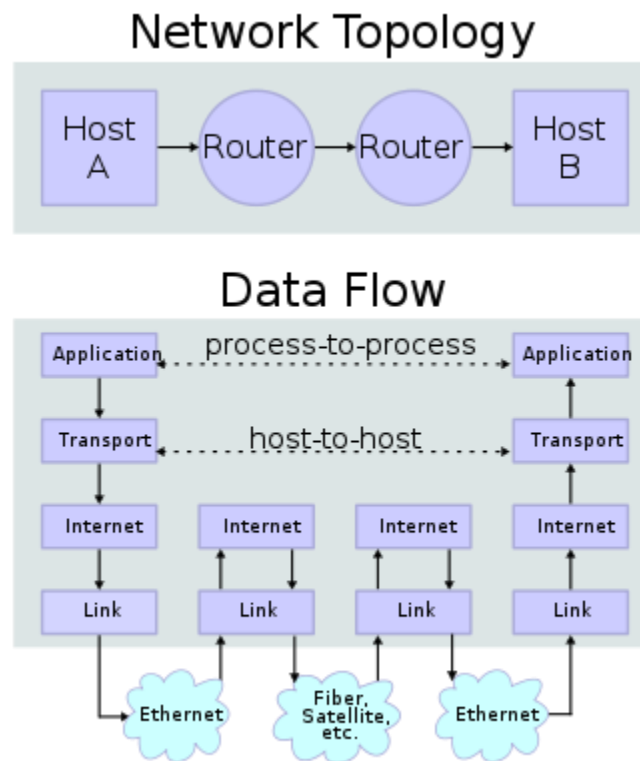


Figure 2.2.1b Routes of packets

2.2.2 Inspection principle

The normal message inspection only analyses the contents of lower four layers, which contain the source IP address, destination IP address, the source IP port, destination IP port and the transmission protocol type. As for the application type, the normal message inspection can only identify it through the port number. However, the illegal application on network can hide or fake port number to avoid inspection and monitor, which resulted in an erosion of the network. Deep Packet Inspection was born for this. It can not only inspect the known protocol on non-standard port (HTTP message on non-80 port), but also non-standard protocol on a known port (data flow of Skype on 80 port). That means the traditional equation between port and application is no longer valid.

As a flow monitoring and control technique based on application layer, DPI has its unique operation sequence. When an IP data packet, a TCP or UDP data flow pass the bandwidth management system that based on DPI, the system can read the load content inside of IP packets in order to reorganize the

[RN_2860] [User behaviour analysis based on Deep Packet Inspection]

application-layer information in the OSI model. Then it can proceed to the next step: obtaining the entire content of application programme and shaping the network traffic according to the definition of system management strategy.

2.2.3 Three main recognition technology

2.2.3.1 *Recognition based on “tagged word”*

Different application protocols have their unique features: specific port, specific character string or specific bit sequence. According to the detection method, the recognition based on “tagged word” can be divided into fixed position matching, flexible position matching and state characteristic matching.

For example, the identification of Bit Torrent protocol follows this formula. There is a number that represent the length of the message before every message. When the network is handshaking, it send “19” first, followed by string “BitTorrent protocol”. Then “19BitTorrent protocol” become the “tagged word” of Bit Torrent.

2.2.3.2 *Recognition at application layer gateway*

The control flow and data flow of some protocols are separated. For these protocols, their data flow has no specific feature. Application layer gateway needs to recognize the control flow firstly, then analyse the control flow through a specific gateway based on its protocol, and finally recognize the corresponding data flow.

For each agreement, there need to be different application layer gateway to analyse it, such as SIP, H323 protocol. SIP/H323 negotiate its data tunnel through signalling interaction. The data flow is RTP format encapsulated voice flow. That is to say, the detection of RTP flow cannot obtain which kind of protocol this RTP flow was established on, unless by analysing the SIP/H323 protocol interaction.

2.2.3.3 *Recognition based on behaviour pattern*

This kind of recognition technique can judge the ongoing or forthcoming behaviour according to existing behaviour of consumer’s terminal. Recognition based on the behaviour pattern is usually used on those operations that cannot be identified through protocols.

For example, the data flows of spam email and normal email are same from the view of the network model. Only by analysing the user behaviour toward different email (white/black list), can the

protocol and flow of spam email be identified.

2.3 Advantage of using DPI on user behaviour analysis

The characteristic of reading into the load content of packet make DPI a possible way to obtain large amount of user behaviour information. Additionally, there is another advantage. If the network carrier use DPI, typically they can get much more comprehensive data than the network service company. Because the network carrier act as a tube and can look inside at every flow of data link as Figure 2.3a shows.

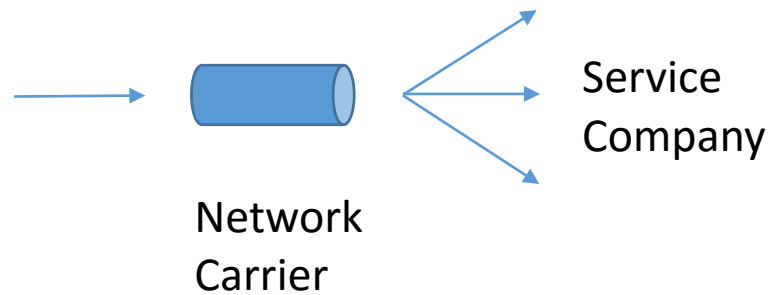


Figure 2.3a

Additionally, the security of data and robustness of inspection system can be guaranteed due to the complex execution flow and multiple fault-tolerant principle of Deep Packet Inspection. Under the rigorous usage management from network carrier and government, the privacy of data would be largely protected.

Chapter 3: Design and Implementation

3.1 nDPI platform deployment and testing

3.1.1 Open source plans selection

Among many Deep Packet Inspection open source implementation plan, two plans were selected and compared: nDPI and Libprotoident. (See table 3.1a)

Table 3.1a

	Application protocols	Applications	Web services
nDPI	15 out of 17 (ssl)	17 out of 22	Better performance
Libprotoident	15 out of 17 (ssl, effective to encryption protocol)	14 out of 22	Normal performance because of only using first bytes

nDPI scheme was chose due to its better performance. It is a library inherited from OpenDPI, maintained by NTOP company. nDPI was published under the permission of General Public License (GPL). Its goal is to increase the number of available protocols and extend the original library. Besides Unix, nDPI also support Windows and can be modified by developers in order to disable some unnecessary characteristics that caused a decline in the efficiency of DPI engine. In this way, nDPI is better for flow control applications. With the contribution of developers, nDPI now support 170 protocols and still exploring its limit.

3.1.2 Deployment in a real network environment

To better inspect on the whole network state in the lab, the nDPI platform was deployed at the switch of the lab network, which control all the data flow through these links. For that most of the user behaviours could be exclusively recognized according to the protocol type in the Application layer, my research mainly focused on the HTML format of video websites.

The whole deployment architecture was showed as figure 3.1b. In this framework, different users are visiting different website to watch the videos through their devices in this network.

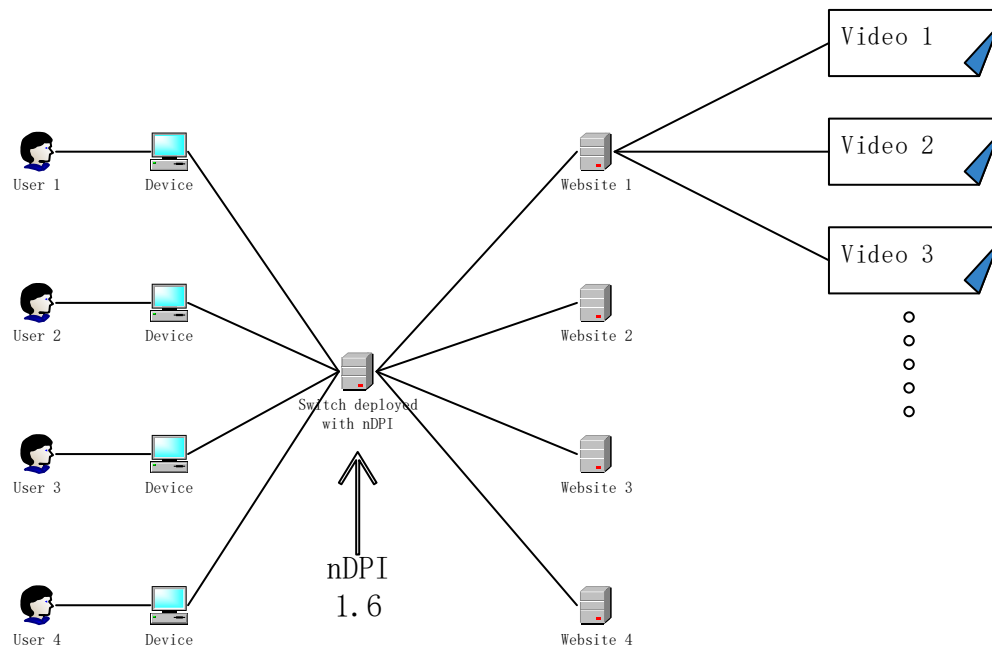


Figure 3.1b

3.2 Real-time grasp the Internet packets and analyses the protocol type

3.2.1 Execution flow

The platform analyses the data packet from the lower layer to the upper layer, which concludes the Data link layer, Network layer, Transport layer and Application layer. See figure 3.2.1a.

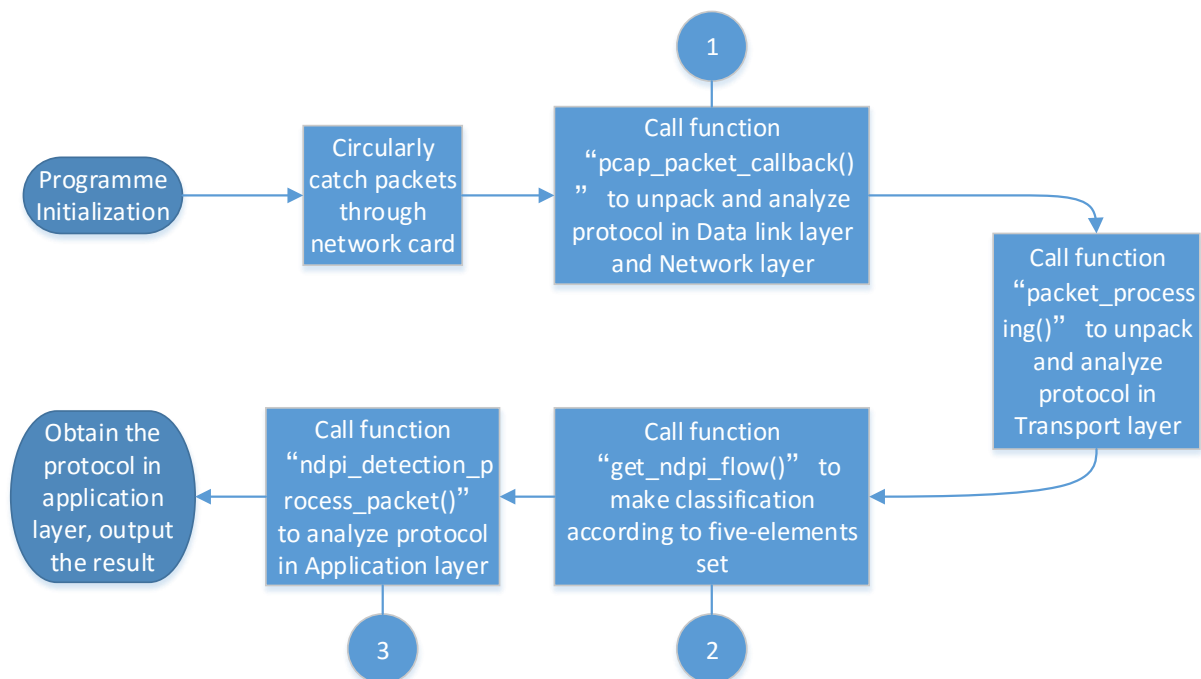


Figure 3.2.1a

Firstly, the platform will call the function “setupDetection()” to initialize programme. Then the system will assign a series of threads to call function “libpcap()” to circularly catch packets through the network card.

For each of the packets, the system will call function “pcap_packet_callback()” to unpack and analyze protocol in Data link layer and Network layer, judge whether it is based on IP or other protocols, and obtain source IP address, protocol type etc. This process can be displayed as a thread as figure 3.2.1b shows.

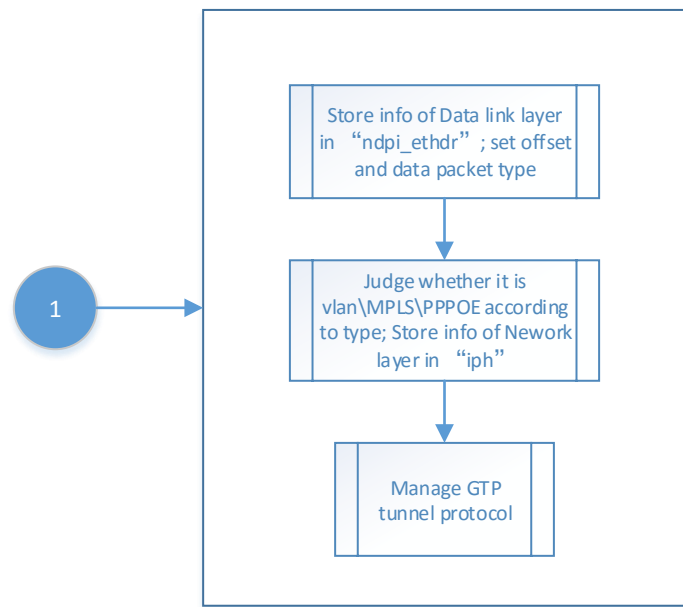


Figure 3.2.1b

After finishing the analysis in the first two layers, the platform will continually call function “packet_processing()” to unpack and analyze protocol in the Transport layer. In this layer, the system call function “get_ndpi_flow()”, which will return the structure: “ndpi_flow” with the transport layer information in it. Finally the system calculates “idx” (index of data flow) based on the five-elements set (source IP address, destination IP address, source port, destination port, protocol type(tcp/udp)). See Figure 3.2.1c.

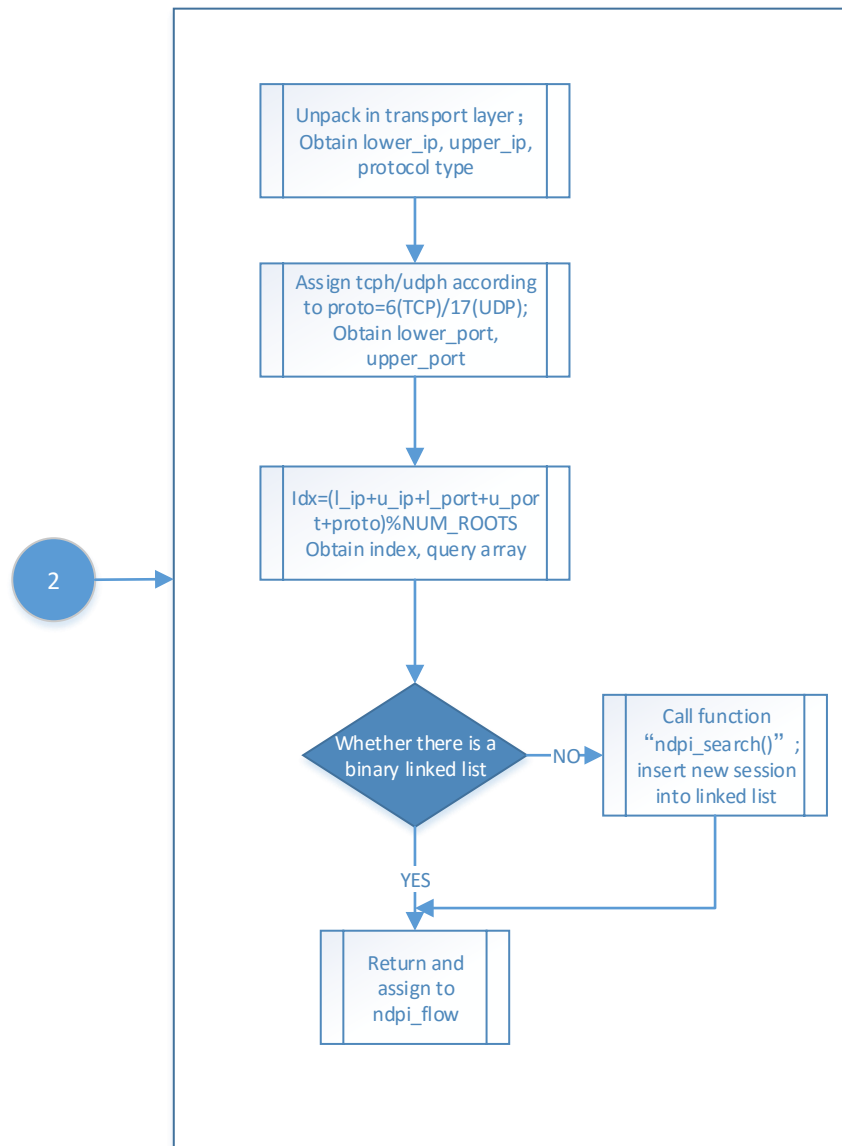


Figure 3.2.1c

The programme will maintain an array that store all the data flow. Variable idx is used to identify those data flows. First the system will calculate the idx based on five-elements set and find whether there is a record on the position of idx of array “ndpi_flows_root[]”. Normally, when the first packet of a data flow try to query the array, it will return with null, then the system will make a new ndpi_flow object. When the system catch the next packets of this data flow and execute the query, it will directly return the existing ndpi_flow constructor, because they both belong to the same flow. The code segment below well demonstrates the function of “idx” variable and what “get_ndpi_flow” return.

```
idx = (vlan_id + lower_ip + upper_ip + iph->protocol + lower_port +  
upper_port) % NUM_ROOTS;  
ret = ndpi_tfind(&flow, &ndpi_thread_info[thread_id].ndpi_flows_root[idx],  
node_cmp);
```

Finally the system will call function “ndpi_detection_process_packet()” to analyze protocol in the Application layer, which is the most important function in protocol analysis. See figure 3.2.1d.

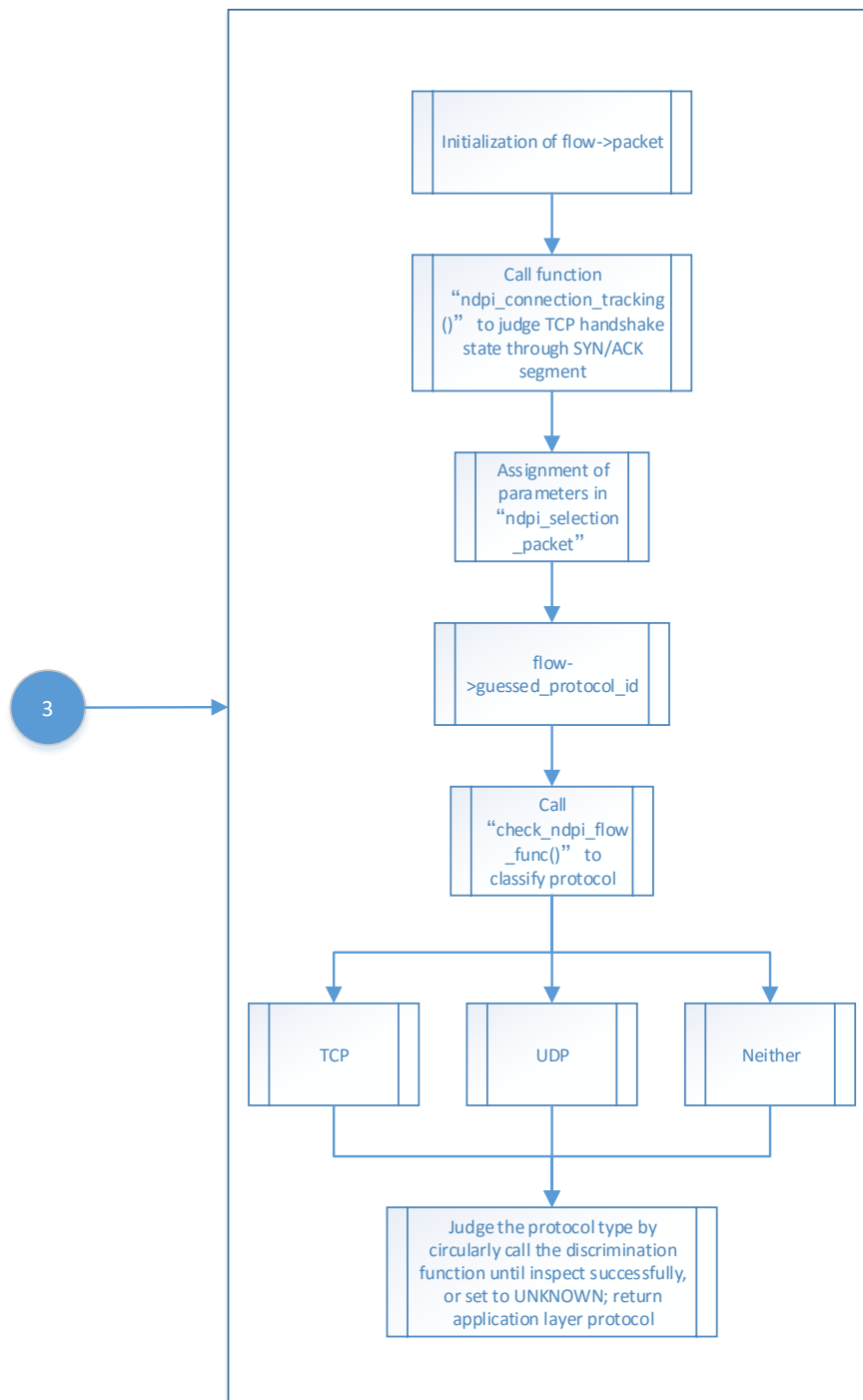


Figure 3.2.1d

The function “ndpi_detection_process_packet()” will firstly initialize the flow->packet constructor. Because, for the same data flow, some variables in the first packet of the flow have been initialized and will not change unless the system inspects a new protocol. However, for each of the data packets, the information in “flow->packet” constructor must be changed.

Then the system will call function “ndpi_connection_tracking()” to judge the position of this packet, for example, the three handshaking state of TCP connection establishment through the value of segments syn, ack, seq and ack_seq.

After that, the constructor “ndpi_selection_packet” will be set, which is used to store the information of lower four layers. This variable belongs to ndpi_selection_bitmask_protocol_size type, whose format is a series of binary bits such as 10101011. And each of the bits correspond to different meaning. The code segment below shows the functions of each bits.

```
#define NDPI_SELECTION_BITMASK_PROTOCOL_SIZE          u_int32_t
#define NDPI_SELECTION_BITMASK_PROTOCOL_IP            (1<<0)
#define NDPI_SELECTION_BITMASK_PROTOCOL_INT_TCP       (1<<1)
#define NDPI_SELECTION_BITMASK_PROTOCOL_INT_UDP       (1<<2)
#define NDPI_SELECTION_BITMASK_PROTOCOL_INT_TCP_OR_UDP (1<<3)
#define NDPI_SELECTION_BITMASK_PROTOCOL_HAS_PAYLOAD   (1<<4)
#define NDPI_SELECTION_BITMASK_PROTOCOL_NO_TCP_RETRANSMISSION (1<<5)
#define NDPI_SELECTION_BITMASK_PROTOCOL_IPV6          (1<<6)
#define NDPI_SELECTION_BITMASK_PROTOCOL_IPV4_OR_IPV6  (1<<7)
#define NDPI_SELECTION_BITMASK_PROTOCOL_COMPLETE_TRAFFIC (1<<8)
```

Finally, the system will call function “guessed_protocol_id” and “check_ndpi_flow_func()” to inspect the protocol type in the application layer. The packets will be distributed to different interfaces according to its protocol in the transport layer (TCP\UDP\neither). The system will judge the protocol type by circularly call the discrimination function until inspection successfully, or set to UNKNOWN. The whole execution flow of the inspection platform is finished. And the platform will return the statistical result of the packets in this pcap file, as figure 3.2.1e shows.

```

Using nDPI (1.6.0--0-) [1 thread(s)]
Reading packets from pcap file test2.pcapng...
Running thread 0...

WARNING: only IPv4/IPv6 packets are supported in this demo (nDPI supports both IPv4 and IPv6), all other packets will be discarded

nDPI Memory statistics:
  nDPI Memory (once):      91.46 KB
  Flow Memory (per flow):  1.92 KB
  Actual Memory:          2.79 MB
  Peak Memory:            2.79 MB

Traffic statistics:
  Ethernet bytes:          9643823      (includes ethernet CRC/IFC/trailer)
  Discarded bytes:         84
  IP packets:              14981        of 14983 packets total
  IP bytes:                9284279      (avg pkt size 619 bytes)
  Unique flows:            528
  TCP Packets:             14284
  UDP Packets:             692
  VLAN Packets:            0
  MPLS Packets:            0
  PPPoE Packets:           0
  Fragmented Packets:      0
  Max Packet size:         1480
  Packet Len < 64:          7433
  Packet Len 64-128:        399
  Packet Len 128-256:       340
  Packet Len 256-1024:     1351
  Packet Len 1024-1500:    5458
  Packet Len > 1500:        0
  nDPI throughput:         1.96 M pps / 9.41 Gb/sec
  Traffic throughput:      288.22 pps / 1.42 Mb/sec
  Traffic duration:        51.977 sec
  Guessed flow protos:     37

Detected protocols:
  Unknown      packets: 1      bytes: 83      flows: 1
  DNS          packets: 565    bytes: 62196   flows: 278
  HTTP         packets: 7974    bytes: 4796498 flows: 157
  SSDP         packets: 43     bytes: 19764   flows: 1
  ICMP         packets: 5      bytes: 2471    flows: 2
  SSL          packets: 4826   bytes: 3337635 flows: 37
  Google       packets: 1363   bytes: 941365  flows: 49
  Apple        packets: 204    bytes: 124267  flows: 3

Protocol statistics:
  Safe          3337635 bytes
  Acceptable    5946561 bytes
  Unrated       83 bytes

```

Figure 3.2.1e

3.2.2 Regular expression summary

To accurately identify and extract the user behaviour information keyword, a survey was made on six dominating video website's HTML code and its format was summarized into Regular Expression.

Regular expression (sometimes called rational expression) is a series of characters that define a search sequence pattern, it is mainly used for pattern string matching. This concept, "Find and replace" operation, appears in the 1950s, when American mathematician Stephen formalized description of regular language, and began to widespread use of Unix text processing tools, an editor, a grep and a filter.

For example, figure 3.2.2a shows the HTML code of Chinese biggest video website, Youku.


```

<!DOCTYPE html>
<html>
<head>
<script charset="utf-8" src="http://static.atm.youku.com/idea/201503/0318/RTB/1/ssp_adx_render_0.1.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>一代宗师-在线播放-《一代宗师》-电影-优酷网，视频高清在线观看</title>
<meta name="title" content="一代宗师">
<meta name="keywords" content="一代宗师">
<meta name="description" content="一代宗师 广东佛山人叶问（梁朝伟 饰），年少时家境优渥，师从咏春拳第三代传人陈华顺学习拳法
<meta name="irTitle" content="一代宗师">
<meta name="irAlbumName" content="一代宗师">
<meta name="irCategory" content="电影">
<link href="http://static.youku.com/v1.0.1112/index/css/youku.css" type="text/css" rel="stylesheet">
<link href="http://static.youku.com/v1.0.1112/v/css/playV5.css" type="text/css" rel="stylesheet">
<script src="http://static.youku.com/v1.0.1112/js/jquery.js"></script>
<script>jQuery.noConflict();</script>
<script src="http://static.youku.com/v1.0.1112/js/prototype.js"></script>

```

Figure 3.2.2a

In this HTML page, the main feature mainly concentrated in head part, the format of the header was summarized below:

```

HTML title:
<title>一代宗师-在线播放-《一代宗师》-电影-优酷网，视频高清在线观看</title>
Subtitle:
<meta name="title" content="一代宗师">
Keyword:
<meta name="keywords" content="一代宗师">
Story and feature:
<meta name="description" content="一代宗师 广东佛山人叶问（梁朝伟 饰），年少时家境
优渥，师从咏春拳第三代传人陈华顺学习拳法，师傅“一条腰带一口气”的告诫，支持他走过兵荒马
乱、朝代更迭的混乱年代。妻子张永...">
Subheading or episode name:
<meta name="irTitle" content="一代宗师">
Album name:
<meta name="irAlbumName" content="一代宗师">
Category(movie or TV series):
<meta name="irCategory" content="电影">

```

Then the regular expression that used to extract the user key word and title can be showed as the code below. The subsequent test showed this method with high sensitivity on the title of videos, which are, combined with the timestamps, the legible indicators of user habits.

```

Extract title:
<title>[\u4e00-\u9fa5]+-在线播放-《[\u4e00-\u9fa5]+》-[\u4e00-\u9fa5]+-优酷
网，视频高清在线观看</title>
Extract subtitle or episode name:
"irTitle.*[\u4e00-\u9fa5]+ "
Extract album name:
"irAlbumName.*[\u4e00-\u9fa5]+"

```

```
Extract category:  
"irCategory.*[\u4e00-\u9fa5]+"
```

3.2.3 Chunked packet encoding format in HTTP message and decompression implementation with C

This is a vital technological point in the implementation of the system. First we need to understand two concepts, data packet and data flow. A data flow may contain a lot of data packets. For example, when we request a webpage, the server will divide the information on the webpage into a number of packets because of the large amount of information. All these data packets belong to a same data flow.

When the client receives those packets, it should reorganize the packets and decompressed them. In this project, the system will use chunked block transmission format, in order to obtain the entire HTTP responding message content. Because we use Content-length to identify the length of the message for the normal HTTP message. However, for some webpage that cannot confirm the length of their message, the system must use chunked encoding format.

Chunk, just as its name implies, was a technique that blocks the large amount of information in to several smaller parts. See figure 3.2.3 a.

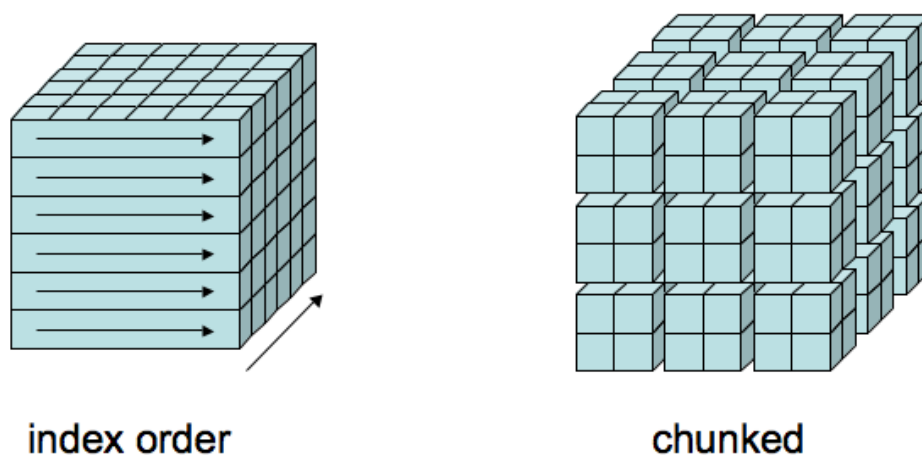


Figure 3.2.3 a.

Chunked encoding consist of several chunks strings and a 0-length chunk as the end mark. For the message that was compressed in gzip format, they were compressed firstly and chunked secondly. So at the client end, we need firstly reorganize the chunks and then decompressed the message as figure 3.2.3b shows.

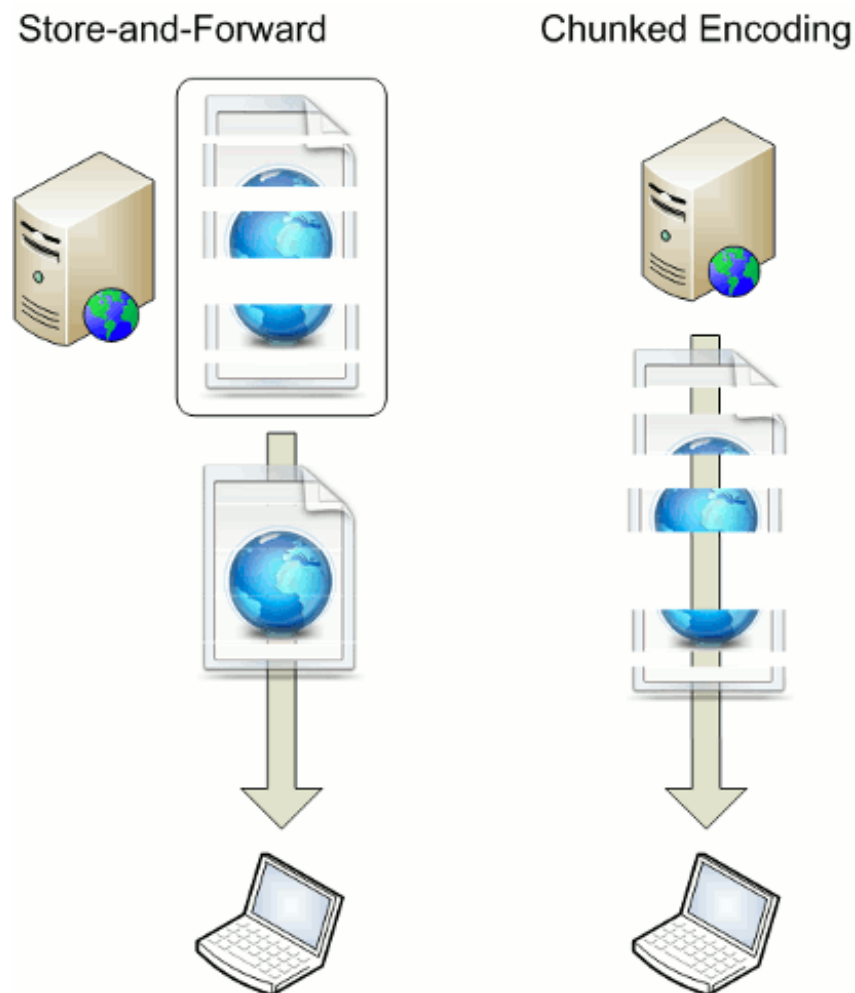


Figure 3.2.3 b.

For example, this is a packet (figure 3.2.3c). The string “transfer-coding” in line3 and line4 indicates it is encoded in chunked format. In line5 “0d 0a 0d 0a” indicate this is the end of the header and will then be the content of the message. “32 35” is the size of the first chunk. In line10, “30 0d 0a 0d 0a” is the end of the chunk string because “30” shows that the current chunk’s size is zero.

1	0000-000F	48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d	HTTP/1.1 200 OK.
2	0010-001F	0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74	.Content-Type: t
3	0020-002F	65 78 74 2f 70 6c 61 69 6e 0d 0a 54 72 61 6e 73	ext/plain..Trans
4	0030-003F	66 65 72 2d 45 6e 63 6f 64 69 6e 67 3a 20 63 68	fer-Encoding: ch
5	0040-004F	75 6e 6b 65 64 0d 0a 0d 0a 32 35 0d 0a 54 68 69	unked...25..Thi
6	0050-005F	73 20 69 73 20 74 68 65 20 64 61 74 61 20 69 6e	s is the data in
7	0060-006F	20 74 68 65 20 66 69 72 73 74 20 63 68 75 6e 6b	the first chunk
8	0070-007F	0d 0a 0d 0a 31 41 0d 0a 61 6e 64 20 74 68 69 731A...and this
9	0080-008F	20 69 73 20 74 68 65 20 73 65 63 6f 6e 64 20 6f	is the second o
10	0090-009F	6e 65 0d 0a 30 0d 0a 0d 0a	ne..0....

Figure 3.2.3 c.

The decompression programme obeys the principle above. It first judge whether the packet is the first one of the HTTP response, and then detect the size of the first chunk to reallocate a part of memory to initialize an array. The system circularly dechunk the packets from the socket until it detects the chunk who shows the current chunk's size is zero.

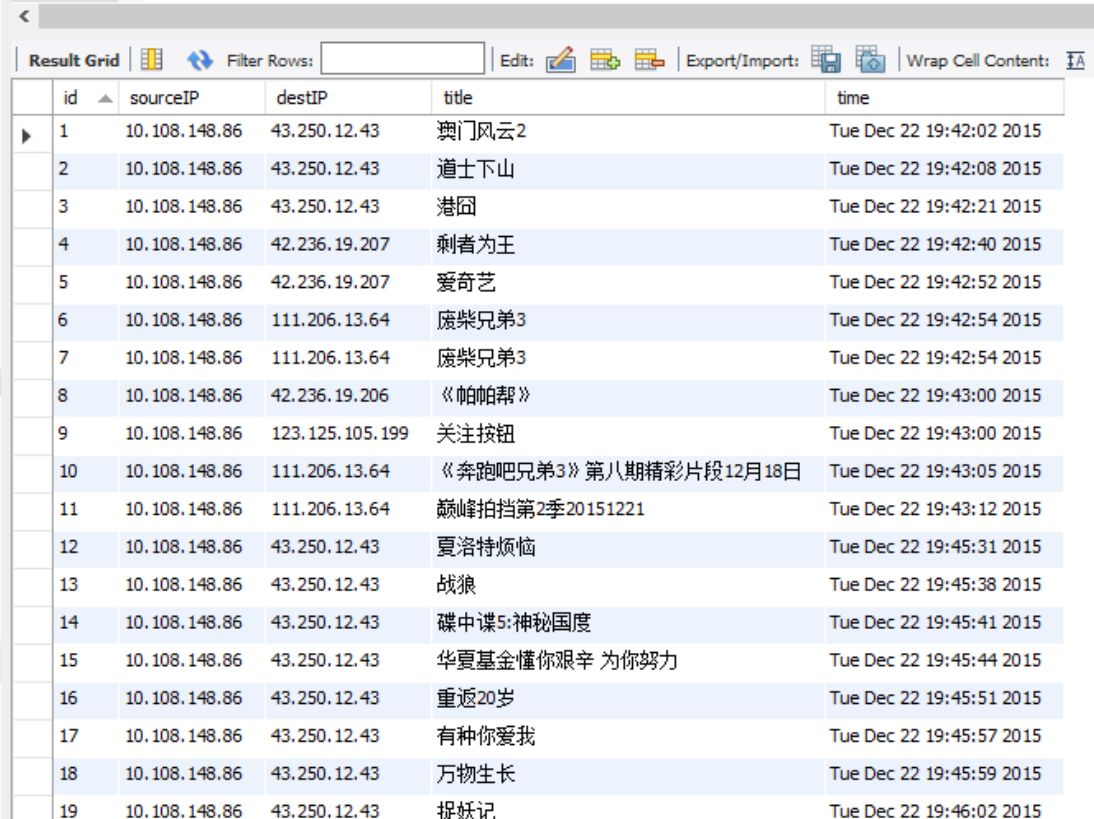
However, there is a problem that this kind of operation will cause a big decline in efficiency due to the reallocation of memory. So I decide to use `inflate()` function in Zlib library to decompress HTTP message that was compressed through Gzip method. This method will directly decompress the message through memory, leaving out the reorganization step that chunked method use.

There is also a disadvantage of the Gzip method. By using Gzip, the correct message can be decompressed out, but only from a part of the whole webpage. Only by catching all the packets of this flow, can the whole page content be shown entirely. However, the subsequent test showed that the information of a video website page normally stored in the first packet. In this way, the platform can not only obtain the necessary information to describe this behaviour, but also save a large amount of efficiency.

3.3 Programming realization of user behaviour analysis system

3.3.1 MySQL storage

After catching and analysing the packets by nDPI system, the user behaviour related factors (source IP address, destination IP address, video title and timestamp) will be filtered in and stored into a MySQL database. See figure 3.3.1a.



id	sourceIP	destIP	title	time
1	10.108.148.86	43.250.12.43	澳门风云2	Tue Dec 22 19:42:02 2015
2	10.108.148.86	43.250.12.43	道士下山	Tue Dec 22 19:42:08 2015
3	10.108.148.86	43.250.12.43	港囧	Tue Dec 22 19:42:21 2015
4	10.108.148.86	42.236.19.207	剩者为王	Tue Dec 22 19:42:40 2015
5	10.108.148.86	42.236.19.207	爱奇艺	Tue Dec 22 19:42:52 2015
6	10.108.148.86	111.206.13.64	废柴兄弟3	Tue Dec 22 19:42:54 2015
7	10.108.148.86	111.206.13.64	废柴兄弟3	Tue Dec 22 19:42:54 2015
8	10.108.148.86	42.236.19.206	《帕帕帮》	Tue Dec 22 19:43:00 2015
9	10.108.148.86	123.125.105.199	关注按钮	Tue Dec 22 19:43:00 2015
10	10.108.148.86	111.206.13.64	《奔跑吧兄弟3》第八期精彩片段12月18日	Tue Dec 22 19:43:05 2015
11	10.108.148.86	111.206.13.64	巅峰拍档第2季20151221	Tue Dec 22 19:43:12 2015
12	10.108.148.86	43.250.12.43	夏洛特烦恼	Tue Dec 22 19:45:31 2015
13	10.108.148.86	43.250.12.43	战狼	Tue Dec 22 19:45:38 2015
14	10.108.148.86	43.250.12.43	碟中谍5:神秘国度	Tue Dec 22 19:45:41 2015
15	10.108.148.86	43.250.12.43	华夏基金懂你艰辛 为你努力	Tue Dec 22 19:45:44 2015
16	10.108.148.86	43.250.12.43	重返20岁	Tue Dec 22 19:45:51 2015
17	10.108.148.86	43.250.12.43	有种你爱我	Tue Dec 22 19:45:57 2015
18	10.108.148.86	43.250.12.43	万物生长	Tue Dec 22 19:45:59 2015
19	10.108.148.86	43.250.12.43	捉妖记	Tue Dec 22 19:46:02 2015

Figure 3.3.1 a.

A table “my_table” was created with five attributes. Every log means a data flow, and also, a webpage. “id” is used to identify every user log uniquely, which is the primary key of this table. “sourceIP” and “destIP” represent the device address of the client and website address of the server, which can be understood as the unique identifier of consumer and video website. “title” is the title of the video (Because the object of my research is Chinese people, most of the video title is in Chinese). “time” is the time when the data flow was transmitted, which means the access time of the user. The database was maintained at localhost and can be connected to the local server.

3.3.2 Front-end query system

Finally, a primary analysis front-end platform was programmed for querying every user's access behaviour details and regional integrated situation. Webpage was selected as a tool of my front-end query system because of its dynamism and transparency. The front-end query system was deployed on Apache Tomcat8.0 and used Spring Framework, which is an application framework and inversion of control container for the Java platform. See figure 3.3.2a.

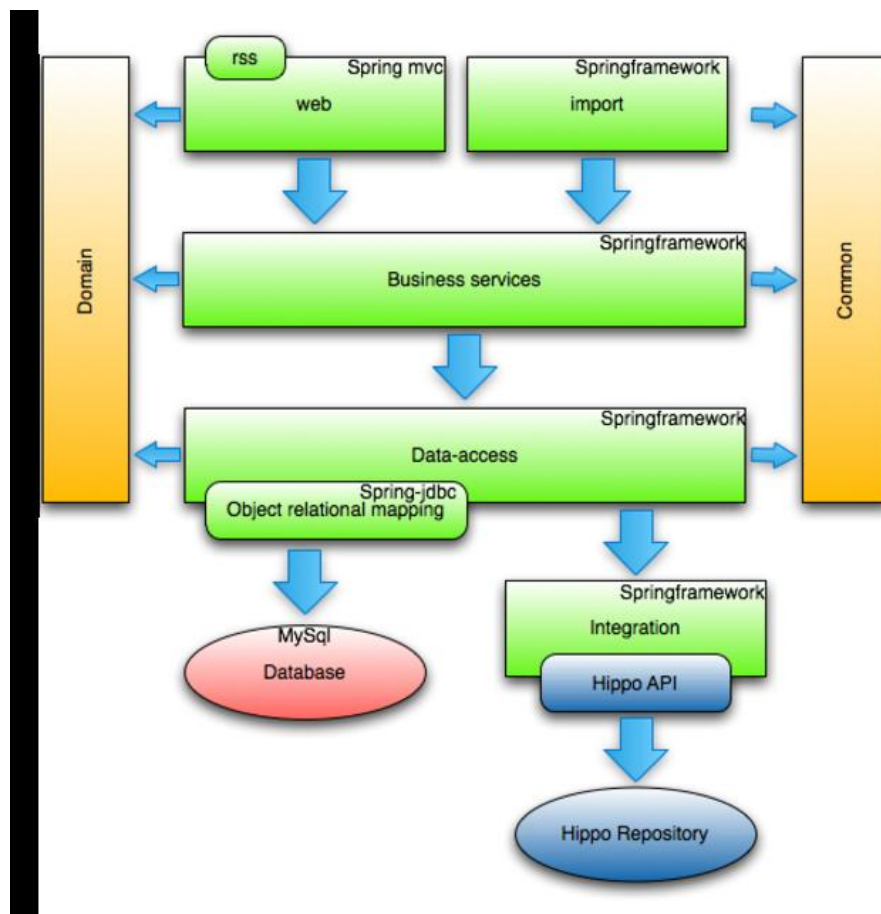


Figure 3.3.2a

By accessing the localhost IP address, the user can enter the user behaviour analysis system that based on DPI as figure 3.3.2b shows.

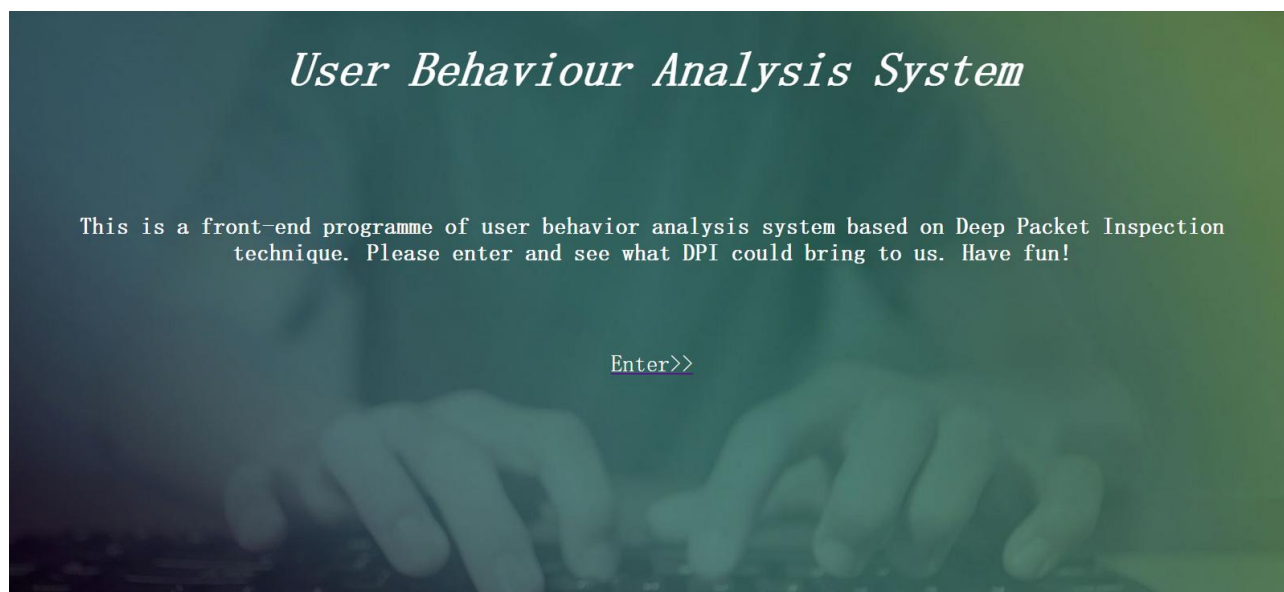


Figure 3.3.2b

[RN_2860] [User behaviour analysis based on Deep Packet Inspection]

The main page was divided into two parts. The upper part is used to let the user set any conditions while querying, which include the query method, log ID, source IP address, key word or phrase of video and time range. The two buttons under the conditions are used to submit the form and reset the form. As soon as the submit button was pressed, the system will make a query to the data base and return the result. The lower part is used to display the query result and the statistical number of eligible logs. The size of each page is set to 10 logs. See figure 3.3.2c.

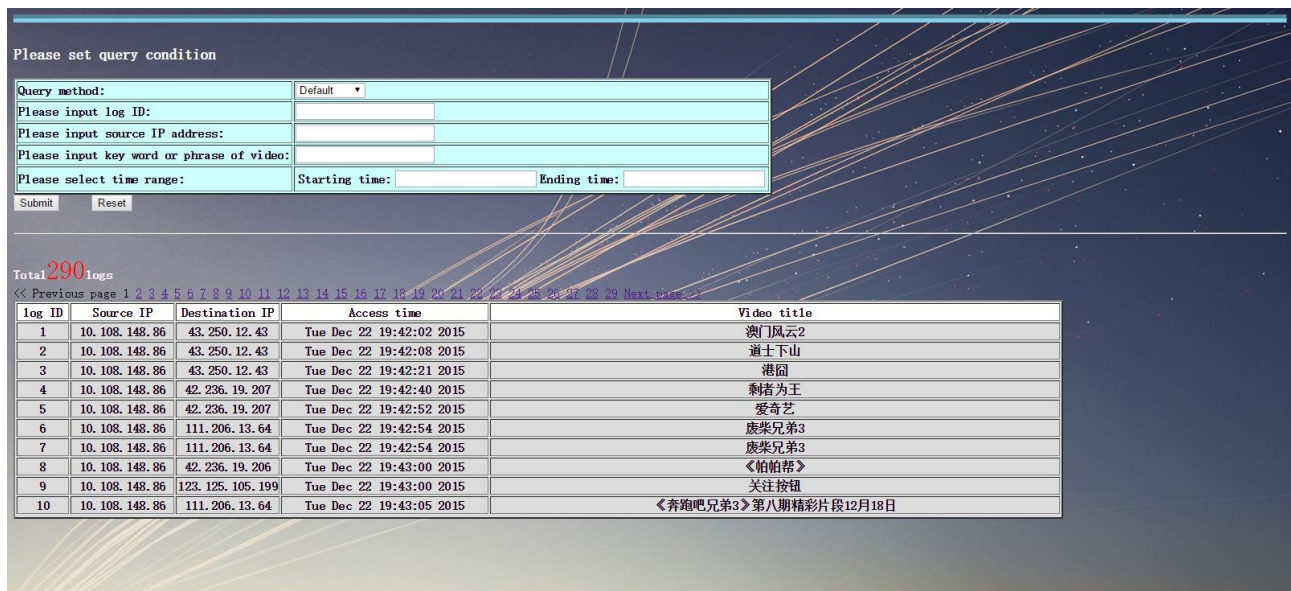


Figure 3.3.2c

The three query method of the platform are “Default”, “Find user” and “Find video”. The “Default” method will give a user the whole authority to change every condition and will return the entire information of each log. See figure 3.3.2d.

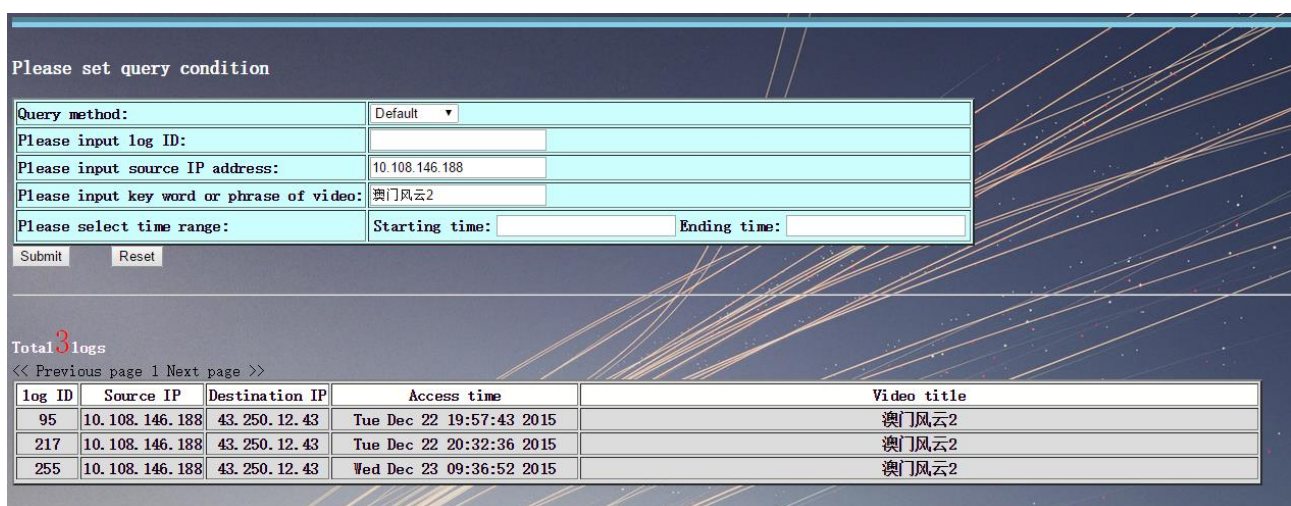


Figure 3.3.2d

In the “Find user” method, users are only allowed to set keyword of the video title and time range. The system will give the source IP address and statistical number of access times of every source IP. This method is used when the user wants to find who and how frequently the audience watch this video, which is “Find user” means. See figure 3.3.2e.

The screenshot displays a web interface for setting query conditions. The title is "Please set query condition". The "Query method:" dropdown is set to "Find user". The "Please input key word or phrase of video:" field contains "秦时明月". The "Please select time range:" section has empty fields for "Starting time:" and "Ending time:". Below the form are "Submit" and "Reset" buttons.

Below the form, the results are displayed. It shows "Total 5 logs" and a pagination link "<< Previous page 1 Next page >>". A table lists the source IP addresses and the number of access times for each.

sourceIP	Number of access times
10.108.146.188	5
86.148.108.10	2
208.151.108.10	2
10.108.148.86	1
130.145.108.10	1

Figure 3.3.2 e

In the “Find video” method, users are only allowed to set the source IP address and time range. The system will give the title of videos and statistical number of access times for every video. This method is used when the user wants to find which video a specific audience like, which is “Find video” means. See figure 3.3.2 f.

Please set query condition

Query method:	Find video ▼	
Please input source IP address:	10.108.146.188	
Please select time range:	Starting time:	Ending time:

Submit Reset

Please set query condition

Query method:	Default ▼	
Please input log ID:		
Please input source IP address:	10.108.146.188	
Please input key word or phrase of video:		
Please select time range:	Starting time:	Ending time:

Submit Reset

Total **77** logs

<< Previous page 1 [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) Next page >>

title	Number of access times
秦时明月 TV版	5
长在面包树上的女人	5
战狼	5
400 Bad Request	4
爹妈满院	4
枪侠	3
家和万事兴 TV版	3
碟中谍5:神秘国度	3
爱情碟中谍 TV版	3
杀破狼2	3

Figure 3.3.2 f

Every time when a query has been finished, the query method will be turned into default automatically for further use.

Chapter 4: Results and Discussion

The outputs of this project contained a nDPI platform on switch, a database and a front-end query system. The first two parts were developer-oriented and can be seen as a black box to the users. White/black box testing was conducted to ensure the robustness and accuracy of the system. Evaluation of the effectiveness and usability of the system were carried out. By comparing the results of the analytic system with the actual user behaviours and habits to verify the reliability, I get the conclusion of that the deep packet inspection will give the expected result and the prospect to be employed in more advanced user behaviour analysis.

For the user, we can divide them into several groups. The video websites and video authors may want to use “Find user” to know the audience of one specific video. They can know what kind of people like to watch this video and when they prefer to watch it, like news in the morning or TV series at night. Even more, when they directional assemble their expected audience, the video website can also analyse user behaviour and tastes through “Find video” with regard to each individuals. Those statistical data can be applied in many aspects as video promotion, accurate advertisements launching and intelligent video recommendation etc. For the network carrier, they can not only inspect the integrated regional network state, but also obtain the valuable user behaviour data. Those can be applied in their user-oriented network product development and more intelligent flow control programme, such as network video aggregation software and dynamic bandwidth distribution. From all consumers’ aspect, user behaviour analysis system based on DPI will be an unprecedented technique that may change our life.

Chapter 5: Conclusion and Further Work

This user behaviour analysis system showed a primary but unprecedented use of Deep Packet Inspection technology. People only use DPI on illegible packets inspection regional flow management in the past. Now I proved that with the help of DPI, the network carrier can collect their own user behaviour information, which can be applied in several aspects like accurate advertisement operation, recommendation based on user interests and habits. However, every technology is a double-edged sword. In the foreseeable future, the problem of privacy invasion and the user data security will be definitely raised. Only DPI technique was applied under the rigorous control by government or associations, can it be used legally and justly.

References

- [1] Luft, S. J., & Chiang, P. (2010). *Network element architecture for deep packet inspection*. US, US 7719966 B2.
- [2] Kumar, S., Dharmapurikar, S., Yu, F., Crowley, P., & Turner, J. (2006). *Algorithms to accelerate multiple regular expressions matching for deep packet inspection*. *Acm Sigcomm Computer Communication Review*, 36(4), 339-350.
- [3] “nDPI execution flow” <http://blog.csdn.net/liuchonge/article/details/50118943>
- [4] Alsbih, A., Janson, T., & Schindelhauer, C. (2011). *Analysis of Peer-to-Peer Traffic and User Behaviour*. *International Conference on Internet Technologies & Applications*.
- [5] Nagy, I. K., & Gasparpapanek, C. (2008). *User behaviour analysis based on time spent on web pages*. *Web Mining Applications in E-commerce and E-services*, 117-136.
- [6] Swaminathan, V., & Wei, S. (2011). *Low latency live video streaming using HTTP chunked encoding*. *IEEE, International Workshop on Multimedia Signal Processing* (pp.1-6).
- [7] 孙卫琴. (2004). *Tomcat 与 Java Web 开发技术详解*. 电子工业出版社.
- [8] Johnson, R., Hoeller, J., Arendsen, A., Risberg, T., & Kopylenko, D. (2006). *Professional java development with the spring framework*. *Apc*, 195-237.
- [9] Lin, P. C., Lin, Y. D., Lai, Y. C., & Lee, T. H. (2008). *Using string matching for deep packet inspection*. *Computer*, 41(4), 23-28.
- [10] Collins, R. T. G. (2009). *Privacy implications of deep packet inspection technology: why the next wave in online advertising shouldn't rock the self-regulatory boat*, the. *Ga.l.rev.*

Acknowledgement

Thanks to my supervisors Pro.Yao and Dr. Fang. They inspire me on my idea of this project. Thanks to my mentors Danyang and Chong. You are the best teachers and friends during my final project. You help me with the deployment of DPI system and help me to fix the technical difficulties I met when I first learn the source code. It is a great honour to work with you. Finally, thanks to my parents and all my friends. It is you who always encourage me during my four-year undergraduate life. Thank you with all my heart!

Risk Assessment

This project aims develop a user behaviour analysis system based on Deep Packet Inspection. After the deployment of nDPI system on the switch board of our lab, I real-time grasped the packets from Internet and filtered and stored user behaviour-related factors into a MySQL database. Then I implement a front-end query system to show the result. There are some factors that may have an impact on the achievement of this project, which I will list below.

Table 1

Description of Risk	Description of Impact	Likelihood Rating	Impact Rating	Preventative Actions
Information stored in second or further packet	Have to reorganize the packets into flow, which decline efficiency	Likely	Serious	Filter those flows and reorganized the packets with Chunked method
System crash	Discontinuity in recording the user behaviour data	Unlikely	Major	Back up the system on another switch and start it when emergency
Server crash	User cannot visit the front-end website	Moderate	Major	Automatically restart the server when crash
Data flooding	Decline the efficiency of data processing	Likely	Serious	Deploy on distributed system to process the data
DDOS attack	Effective approaches will be hidden in fake approaches	Rare	Serious	Use safe and static website back-end security solution
SQL injection	Error in query actions	Rare	Serious	Check user's each inputs with regular expression

Environmental Impact Assessment

The only one impact of this project is about privacy invasion resulted from the use of log data. This problem also exists in other industries that use the user behaviour analysis techniques. With my perspective, there are three principles the user of this product must obey. First, the ownership of all the collected data must belong to the user themselves, instead of the companies. The ownership of data should be protected just as the ownership of property. Second, the companies who use the data to provide information service should store and transport the data in security, which is the responsibility of the enterprise. Third, the companies must give the user the right to know and choose whether they want their data be used prior to any operations. Every technique is a double edged sword, only it was applied under the rigorous control by government or associations, can it be used legally and justly.

北京邮电大学
本科毕业设计（论文）任务书
Project Specification Form

学院 School	International School	专业 Programme	Telecommunications	班级Class	2012215104
学生姓名 Name	QIAN Cheng	学号 BUPT student no	2012212860	学号 QM student no	120721267
设计（论文）编号 Project No.	RN_2860				
设计（论文）题目 Project Title	User behavior analysis based on DPI(Deep Packet Inspection)				
论文题目（中文）	基于深度包检测的用户行为分析				
题目分类 Scope	Research	Networks	Simulation		

主要任务及目标Main tasks and target:	By
Task 1: Understand the principle of deep packet inspection	30 December 2015
Task 2: Analysis of the open source code--NDPI	01 March 2016
Task 3: Analysis of user behavior and habits based-on the captured packets	01 May 2016
Task 4:	null

Measurable outcomes
1) The NDPI platform deployment and testing
2) Real-time grasping the Internet packets and analyzes the protocol type
3) Programming realization of user behavior analysis system

主要内容Project description:

DPI is a software solution that monitors a network's data stream and identifies protocols and applications, inappropriate URLs, intrusion attempts and malware by looking deep into data packets. DPI provides important security and translation functions by inspecting incoming packets, reassembling and decompressing them, analyzing the code and passing data to appropriate applications and services. We can analyze the user behavior and habits by DPI.

Project outline

Handle NDPI working structure;
Deeply inspect all layers of the packet with NDPI source code in C;
Program to implement reorganization of packets and data stream;
Grasp real-time Internet packets by NDPI and write condition to filter out the useless packets;
Focus on research and analysis on HTTP message in application layer and gain the useful information for user behavior analysis;
Store user information and behavior-related factors (e.g. IP, URL, timestamp and accessing page content) into MySQL database;
Conduct simple statistical analysis and data mining on user behavior details;
Program front-end platform to generate user portrait, for querying every user's access behavior details and regional integrated situation;
Organized results and write paper.

Fill in the sub-tasks and select the cells to show the extent of each task

	Nov	Dec	Jan	Feb	Mar	Apr	May
Task 1: Understand the principle of deep packet inspection							
Handle NDPI working structure;							
Deeply inspect all layers of the packet with NDPI source code in C;							
Task 2: Analysis of the open source code--NDPI							
Program to implement reorganization of packets and data stream;							
Grasp real-time Internet packets by NDPI and write condition to filter out the useless packets;							
Focus on research and analysis on HTTP message in application layer and gain the useful information for user behavior							
Task 3: Analysis of user behavior and habits based-on the captured packets							
Store user information and behavior-related factors (IP, URL, timestamp and accessing page content) into MySQL database;							
Conduct simple statistical analysis and data mining on user behavior details;							
Program front-end platform to generate user portrait, for querying every user's access behavior details and regional integrated situation;							
Organized results and write paper.							
Task 4:							

北京邮电大学
BBC6521 Project 毕业设计 2015/16

Early-term Progress Report
初期进度报告

学院 School	International School	专业 Programme	Telecommunication Engineering and Management	班级 Class	2012215104
学生姓名 Student Name	Cheng Qian	学号 BUPT Student No.	2012212860	学号 QM Student No.	120721267
设计（论文）编 号 Project No.	RN_2860	电子邮件 Email	tom10ye@bupt.edu.cn		
设计（论文）题 目 Project Title	User behavior analysis based on DPI(Deep Packet Inspection)				
<p>已完成工作： Finished Work:</p> <p>Regular meeting was arranged with my supervisor on every Tuesday. With the help from Professor Yao, I made a schedule of the whole project and checked our milestones every week, which guaranteed the progress of the project. Until now, I have deployed a DPI platform of version nDPI1.6 on Ubuntu system and successfully real-time grasped the Internet packets and analyzed the protocol type in Data Link layer, Network layer, Transport layer and Application layer. Due to the open nature of Deep Packet Inspection technology, I read a lot of blogs on CSDN forum to understand the principle and working structure of the nDPI model. Then I analyzed its source code in C with GDB test. Tracking the change of parameters and functions, I handled the basic execution flow of the code. For that most of the user behaviors could be exclusively recognized according to the protocol type in Application layer, my research mainly focused on the HTML format of video websites. To accurately identify and extract the user behavior information keyword, I made a survey on six dominating video website's HTML code and summarize its format into Regular Expression. The subsequent test showed this method with high sensitivity on the categories of videos, which are, combined with the timestamps, the legible indicators of user habits. The problem I am facing is the reorganization of packets and data stream after transmission in the network. It required an analysis on the special fields of IP packet header and TCP header, especially the Fragment Offset in IP and SYN/ACK segment in TCP and also the message protocol in HTTP. The solution I was working on is to allocate a big part of memory for initializing an array to sort the packets, but the efficiency is enormously declined. And I am still exploring the optimized method.</p>					
是否符合进度？ On schedule as per GANTT chart?			[YES]		

下一步:

Next steps:

Firstly, I am going to store user information and behavior-related factors (IP, URL, timestamp and accessing page content) into a MySQL database. Then I will conduct simple statistical analysis and data mining on user behavior details. After fully conversant with the data format, I plan to program a primary analysis front-end platform to generate user portrait, for querying every user's access behavior details and regional integrated situation. And I will also organize the results and write a part of the final paper.

北京邮电大学
本科毕业设计（论文）中期进展情况检查表
Mid Term Check Form

学院 School	International School	专业 Programme	Telecommunications w	班级 Class	2012215104
学生姓名 Name	QIAN Cheng	学号 BUPT student no	2012212860	学号QM student no.	120721267
设计（论文）编号 Project No.	RN_2860				
设计（论文）题目 Project Title	User behavior analysis based on DPI (Deep Packet Inspection)				
题目分类 Scope	Research	Networks	Simulation		

主要内容：（毕业设计（论文）进展情况，字数一般不少于1000字）

Main body: The progress of the research on the project. Total number of words is no less than 1000.

目标任务 Targets set at initiation	Deployment of NDPI platform; A programme to implement reorganization of packets and data stream; A database of user behavior information; A primary analysis front-end program; A part of final paper.
是否完成目标 Targets met? Yes/No	Yes
目前已完成任务 Finished Work	<p>Until the midterm check session, I will have finished all the expected midterm task. Regular meeting was arranged with my supervisor on every Tuesday. With the help from Professor Yao, I made a schedule of the whole project and checked our milestones every week, which guaranteed the progress of the project.</p> <p>Until now, I have deployed a DPI platform of version nDPI1.6 on Ubuntu system and successfully real-time grasped the Internet packets and analyzed the protocol type in Data Link layer, Network layer, Transport layer and Application layer.</p> <p>Due to the open nature of Deep Packet Inspection technology, I read a lot of blogs on CSDN forum to understand the principle and working structure of the nDPI model. I have read <pcapreader-source code analysis>, <registration and maintenance of protocols> and <FTP deep packet inspection>.</p> <p>Then I analyzed its source code in C with GDB test. Tracking the change of parameters and functions, I handled the basic execution flow of the code. For that most of the user behaviors could be exclusively recognized according to the protocol type in Application layer, my research mainly focused on the HTML format of video websites.</p> <p>To accurately identify and extract the user behavior information keyword, I made a survey on six dominating video website's HTML code and summarize its format into Regular Expression. The subsequent test showed this method with high sensitivity on the categories of videos, which are, combined with the timestamps, the legible indicators of user habits.</p> <p>Then I have stored part of user information and behavior-related factors (IP, URL, timestamp and accessing page content) into a database. I am managing to make a experimental demo of conducting simple statistical analysis and data mining on user behavior details.</p>

<p>尚需完成的任务 Work to do</p>	<p>The expected date of finishing all the rest part of the whole project will be approximately on May 10th. After fully conversant with the data format, I plan to make a research on several data mining algorithms and select one that is most suitable for our data format and user behavior analysis system. Continuing work will be programming a primary analysis front-end platform to generate user portrait, for querying every user's access behavior details and regional integrated situation. The majority of the rest time will be spent on white/black box testing to ensure the robustness and accuracy of the system. Evaluation of the effectiveness and usability of the system will be carried out. I will compare the results of the analytic system with the actual user behaviors and habits to verify the reliability. Then I will get the conclusion of whether the deep packet inspection will give the expected result and the extent to be employed in user behavior analysis. After that I will real-time collect the packets of a period of time and display the result, which is for the final demo and viva. Finally, I will organize the results and write the entire final paper. Because of the characteristic of my project is research, I might provide some possible applications but I am not going to implement that. For those ideas, I will also write into the additional works in my final paper.</p>	
<p>存在问题和解决办法 Problems and Solutions</p>	<p>存在问题 Problems</p>	<p>The problem I am facing is the reorganization of packets and data stream after transmission in the network. It required an analysis on the special fields of IP packet header and TCP header, especially the Fragment Offset in IP and SYN/ACK segment in TCP and also the message protocol in HTTP. Another question is that what we can get from the header of HTML code is just the name and category (film or TV series) of the video, instead of the type (romantic or realistic), which make it difficult to detect the user's preference on video.</p>
	<p>拟采取的办法 Solutions</p>	<p>The solution I was working on is to allocate a big part of memory for initializing an array to sort the packets, but the efficiency is enormously declined. And I am still exploring the optimized method. For the second problem, I pictured to connect the user behavior database with an external open source database filled with video details information to make a more comprehensive analytics.</p>
<p>最终论文结构 Structure of the final report</p>	<p>Specification; Abstract (Both in English and Chinese);A short overview of the whole part. Keywords; Table of contents;</p> <p>CHAPTER 1: Introduction;</p> <p>CHAPTER 2: Motivation and background; Briefly describe the project; Highlight the creative point and field; Schedule of the project and show that I have met the aims stated in the specification.</p> <p>CHAPTER 3: Design and implementation; Introduction of Deep Packet Inspection technology and my design of User Behavior Analysis system; The working process of DPI platform of version nDPI1.6 on Ubuntu system and how it real-time grasped the Internet packets and analyzed the protocol type in Data Link layer, Network layer, Transport layer and Application layer; Principle of extract user behavior-related information from network packets; A survey on six dominating video website' s HTML code and summarize its format into Regular Expression; Several possible data mining algorithms suitable for user behavior analysis. Proposed analysis model and scheme; Implementation of the analysis system;</p> <p>CHAPTER 4: Results and discussion; Evaluation of the effectiveness and usability of the system;</p>	

	<p>Comparison of the results of the analytic system with the real user behaviors and habits to verify the reliability; The problem I met and how I resolved;</p> <p>CHAPTER 5: Conclusion,a clear summary of the design and result; Further work, the discussion of extending applications and several solutions to improve the efficiency and automation of the system;</p> <p>References; Acknowledgements; Appendices; Some annotation and explanation important code fragments and functions; Risk assessment, the robustness and accuracy of the system, privacy infringement discussion; Environmental impact assessment.</p> <p>(Approximately 50 pages)</p>
日期 Date	06/03/2016

Fill in the sub-tasks and select the cells to show the extent of each task

	Nov	Dec	Jan	Feb	Mar	Apr	May
Task 1: Understand the principle of deep packet inspection							
Handle NDPI working structure;							
Deeply inspect all layers of the packet with NDPI source code in C;							
Task 2: Analysis of the open source code--NDPI							
Program to implement reorganization of packets and data stream;							
Grasp real-time Internet packets by NDPI and write condition to filter out the useless packets;							
Focus on research and analysis on HTTP message in application layer and gain the useful information for user behavior							
Task 3: Analysis of user behavior and habits based-on the captured packets							
Store user information and behavior-related factors (IP, URL, timestamp and accessing page content) into MySQL database;							
Conduct simple statistical analysis and data mining on user behavior details;							
Program front-end platform to generate user portrait, for querying every user's access behavior details and regional integrated situation;							
Organized results and write paper.							
Task 4:							