Interim Report

Topic: Utilising Smartphone Application Data
Student: Wong San Yu (3035273022)
Supervisor: Dr T.W. Chim
Submission Date: 20th January, 2019
Abstract

Waiting for buses have become easier with bus companies introducing their own smartphone applications (apps) for passengers to check when do the buses arrive. However, apps from different companies do not read each other’s data so that passengers have to open two or more apps for checking bus arrival times. That could lead to undesired results such as missing a bus when opening another app to check arrival time. This paper describes the early research stage of a tool that integrates bus arrival time data from different bus companies. The goal of the tool is to retrieve arrival time data from different servers and show them to user in an all-in-one manner. In the current stage, it is now possible to retrieve data from Kowloon Motor Bus (KMB)’s server without using the company’s app, by directly communicating with the server using APIs. The logic of retrieving ETA from New World First Bus and Citybus is also found. Next step is to discover a working method to directly retrieve NWFB/Citybus’s bus arrival times. In the future, this tool might grow into an app which gathers all available public transport arrival time data such that it can make users’ life easier.
# Table of Contents

Abstract ........................................................................................................................................... 2

List of Figures .................................................................................................................................. 4

1 Introduction .................................................................................................................................. 5
  1.1 Current Problem ......................................................................................................................... 5
  1.2 Project Background .................................................................................................................... 5
  1.3 Related Works ............................................................................................................................ 6
  1.4 Route Map of the Project .......................................................................................................... 7
  1.5 Outline of the Progress Report .................................................................................................. 7
  1.6 Project Objective ....................................................................................................................... 7
  1.7 Scope ......................................................................................................................................... 8

2 Methodology ................................................................................................................................... 8
  2.1 Network Sniffing ....................................................................................................................... 8
    2.1.1 Sniffing on KMB’s ETA Data ............................................................................................. 9
    2.1.2 Sniffing on NWFB/Citybus’s ETA Data ............................................................................ 11
  2.2 Reverse Engineering ................................................................................................................... 11

3 Current Progress ............................................................................................................................ 12
  3.1 Work Done on Kowloon Motor Bus’s Application ..................................................................... 12
    3.1.1 Testing the Results from KMB Sniffing ............................................................................. 14
  3.2 Work Done on Citybus/NWFB’s Website ................................................................................... 16
    3.2.1 Breakdown of NWFB/Citybus’s Website .......................................................................... 17
    3.2.2 Testing the APIs from NWFB/Citybus Website ................................................................. 20
  3.3 Difficulties ................................................................................................................................. 22

4 Challenges ...................................................................................................................................... 23

5 Conclusion ..................................................................................................................................... 23

6 Future Plans .................................................................................................................................. 24
  6.1 Continue Work on Citybus/NWFB app and website ................................................................. 24
  6.2 The Showcase Application ......................................................................................................... 24

References ......................................................................................................................................... 25
List of Figures

Figure 1. Users have to switch between NWFB (left) and KMB (right)................................. 5
Figure 2. Bus companies use standalone apps to provide ETA data........................................ 6
Figure 3. The KMB ETA searching site (left) and Web Inspector (right)................................. 9
Figure 4. The Android virtual machine (left) and Wireshark interface (right)....................... 10
Figure 5. Developer Tools in web browser was used to sniff NWFB/Citybus's website........... 11
Figure 6. Try to sniff on the packets sent to and received from KMB server. The right pane in
the Web Inspector displays the request and response information........................................ 12
Figure 7. API for retrieving bus stops in a route with bound specified................................. 12
Figure 8. Sniffing result of the KMB app using Wireshark, showing the API for retrieving
actual ETA ............................................................................................................................. 13
Figure 9. Retrieving all stops for 1A towards Tsim Sha Tsui..................................................... 14
Figure 10. Showing service types for 1A Tsim Sha Tsui bound.............................................. 15
Figure 11. Retrieving actual ETA at stop no.7 for 1A Tsim Sha Tsui bound............................. 15
Figure 12. The NWFB/Citybus website (left) and corresponding sniffing results (right) ......... 16
Figure 13. Rendered response from server after searching for route....................................... 17
Figure 14. Routing variation is shown...................................................................................... 17
Figure 15. The page shown to user (left) and the actual response from server (right).......... 19
Figure 16. Picking a random previous cookie ......................................................................... 20
Figure 17. Retrieving directions of route B5............................................................................. 20
Figure 18. Routing was found for B5 HZMB direction........................................................... 21
Figure 19. Response from server after searching B5 normal routing.................................... 21
Figure 20. The response was not "OK". That indicated the query was not successful............. 22
Figure 21. ETA could not be shown. Query failed................................................................. 22
1 Introduction

Smartphone usage have become more and more widespread in the past decade. Many smartphone applications are popular among users because they provide useful data. However, most of the companies do not provide open data and it is not convenient to get data from several applications in a short time. This project aims at relieving this situation by exploring methods to make data retrieval from multiple applications easier. In the project, the focus will be on data of Estimated Time of Arrival of buses, as it is a problem often faced by general public.

1.1 Current Problem

The everyday commute has always been a problem for working people and students. In the past, passengers have to wait for unknown time at stations for buses. With the introduction of Estimated Time of Arrival (ETA) of buses in bus companies’ apps, commuters know when does the bus come in advance and that helps in time management. However, people often have to enquire ETAs from different apps for different bus routes. Figure 1 serves an example. A user checks ETA for a cross harbor route jointly operated by KMB and New World First Bus. Then the NWFB app (left) says it is now non-NWFB hours. Therefore the user has to open KMB app (right) and check ETA again. Changing apps back and forth could be annoying. A universal tool gathering ETAs from different bus companies will be favourable for commuters.

![Figure 1. Users have to switch between NWFB (left) and KMB (right)](image)

1.2 Project Background

Although users can get ETAs from the bus companies’ apps, the data are not open to public. Bus companies claim that the ETA data are their private property [2] so that they will not provide any means (e.g. tools accessing real time data and/or static data) for users to retrieve ETA data other than using their app. However, the bus company apps are often slow and in some sense not very user friendly. Figure 2 shows three different apps from three bus companies. They have different layouts and UI/UX. Users have to learn how to use the three apps, which is not user-friendly. As a heavy user of bus services, a transportation enthusiast
and CS student, the abovementioned situation of having to switch apps always appear in daily life. Therefore this project is an interesting one which is a way to put knowledge gained in university into practice and benefit the general public.

![Figure 2. Bus companies use standalone apps to provide ETA data.](image)

1.3 Related Works

There are few integrated transportation applications available on the market. They provide one-stop route suggestions and mixed real time and non-real time ETA. However, as a user, non-real time ETA is not preferred. Following is a general comparison of those apps:

  
  Google Maps is a popular way-finding app which provides route suggestions and basic schedules of bus routes and MTR lines. However, the information provided is often not accurate and it does not provide any real time ETA data for the user.

- **Citymapper** ([https://citymapper.com/hong-kong](https://citymapper.com/hong-kong))
  
  Citymapper provides route finding function in several locations over the world including Hong Kong. It integrates most of the public transport in Hong Kong including red minibuses into their service. However, users can only get real time ETA for KMB and LWB routes. For other bus companies, the app still provides ETA but they are only calculated based on the official schedule from bus companies. The calculated ETA are not accurate.
- Moovit (https://moovit.com)
  Moovit is a service similar to Citymapper. Moovit also provides route suggestions for most kinds of public transportation in Hong Kong. It also supports push notifications to alert users on incidents. The ETA system is the same as Citymapper with only KMB and LWB routes having real time ETA.

This project is not going to compete with the above apps, but try to improve their weakness, which is not providing real time ETA for companies other than KMB and LWB.

1.4 Route Map of the Project

For this project, the main point is to explore how do the existing applications get ETA data from bus companies. Therefore, the first thing to do will be studying the behaviour of the applications. After that, the focus will turn to finding techniques to stimulate the applications and integrating the techniques into the final product of this project.

1.5 Outline of the Progress Report

This report proceeds as follows. First, the methodology is presented in chapter 2. That enables better understanding of the network sniffing and reverse engineering techniques being used in the project. Then some findings, that is, the method to get ETA data from KMB’s server and the main logic of retrieving ETA data from NWFB/Citybus’s server, are presented in chapter 3 so that they show the current progress. We will then discuss some difficulties encountered and talk about upcoming plans in chapters 4 and 7.

1.6 Project Objective

This project aims at finding ways to centralise the process of ETA data request and retrieval from different bus companies within one single application. As the ETA data are not public, different approaches have to be tried in order to retrieve data from companies’ servers. If the way is found, a platform will be built for users to view real time bus ETAs retrieved from different companies, eliminating the need of switching apps to get real time ETA. This will be a big step forward as this will be a new channel for users to get required data easily, which makes their life easier.
1.7 Scope

This project manipulates ETA data from bus companies’ apps and combines them into one single application. As the focus shall be on the techniques to get data, not the breadth of data, only ETA data from Kowloon Motor Bus (KMB), Long Win Bus (LWB), New World First Bus (NWFB) and Citybus (CTB) will be covered as there are actually only two data sources. If time allows, ETA data from New Lantao Bus (NLB) will also be covered. The main target of this project is to develop a tool for checking ETA easily. The product will focus more on functionality than the user interface and user experience.

2 Methodology

For this project, the underlying request/response cycle of bus companies’ apps have to be understood first with the help of some existing tools. There are two main approaches to understand the process, namely Network Sniffing and Reverse Engineering.

2.1 Network Sniffing

In the early stage of the project, network sniffing is first tried. Network Sniffing means studying the real time activity of a computer network [3]. The apps must send and receive data packets to/from bus company servers in order to get the ETA data. If the content of the data transmission can be seen, we can then study the logic behind the connections, send similar requests to bus companies and get the ETA data needed.

For network sniffing, two major tools are used. First one is the Developer Tool in browsers and the other is the software Wireshark. Developer Tool built in the network browsers such as Apple Safari and Google Chrome provide simple but necessary tools for users to study network activity. For example, users can view what resources have been downloaded and which request sent to server they correspond to. The Developer Tool is used to do simple checks on network activity from the web browser. Developer Tool is used because no separate installation is needed and it provide essential functionality. User can quickly analyse network activity relating to web pages.

The software Wireshark is used when we are not working in a web browser. Wireshark is a free open source software for analysing data packets in a computer network [4]. Wireshark monitors all the network activity going in and out of the network interface card, where the computer physically connect to the internet. With the help of Wireshark, data packets
transmitted by apps can then be traced. Contents and destinations of the packets can then be explored and used in future work.

The sniffing on smartphone apps runs on a computer with a virtual machine running Android system. Android is chosen because it is free and is very easy to obtain a copy. Bus companies’ apps are installed in the emulated Android environment and their network activity will be logged by Wireshark on the host machine. The logged data can then be studied later.

2.1.1 Sniffing on KMB’s ETA Data

Sniffing on KMB’s data is done. As the KMB app works with an offline database of bus routes and stops, we first tried sniffing on the web page provided by KMB for passengers to check ETA data to check how to obtain those data. Sniffing was done using built-in Developer Tool in Apple Safari (called Web Inspector) for simplicity (see Figure 3, which shows the interface of Web Inspector). When actions were done on the web page, e.g. entering desired route to check ETA, choosing the bus stop you wish to get ETA, the Tool logged what request had been sent to the server and what did the server returned. From sniffing on the website, the important function for querying bus stop list of a route was found.

Figure 3. The KMB ETA searching site (left) and Web Inspector (right)
The function for querying actual ETA was also found, but it sent HTTP POST requests to server which the contents were not readable as it is the nature of POST requests. Therefore sniffing has to be applied also on the KMB app (see Figure 4, which shows the sniffing on KMB app) to investigate more on how to get ETA. By recording the network activity as soon as ETA was queried in the KMB app, the remaining function of retrieving actual ETA was found.

Figure 4. The Android virtual machine (left) and Wireshark interface (right)
2.1.2 Sniffing on NWFB/Citybus’s ETA Data

Sniffing NWFB/Citybus’s data was done in a similar way of what we have done on KMB’s one. Developer Tools in web browser (see Figure 5) was used and the requests and responses were examined in the Developer Tools window. It was different from KMB’s one because all the query parameters were very explicit and values required could be found in the responses by reading the URL.

![Figure 5. Developer Tools in web browser was used to sniff NWFB/Citybus's website](image)

2.2 Reverse Engineering

If network sniffing does not work, or the results gathered are not sufficient, reverse engineering will be trialed. Reverse Engineering means cracking down the application back to lines of code. Android applications, in the form of APK files, are in fact simple compressed ZIP files containing binary codes. Binary codes can then be converted back to source codes for modification. With the source code, the ultimate logic of the app can be understood clearer. Non-public APIs might also be found. With the API, we can directly contact with bus companies’ servers. Binary codes can be converted to different other forms using tool like [apktool](https://github.com/apktool/apktool) and [dex2jar](https://github.com/aliw/dex2jar) as they are free tools and provide sufficient functionalities. After understanding the program logic, the ETA data request/retrieve process can be emulated for later use.
3 Current Progress

In this section, the project progress, up to the submission of this report, will be discussed.

3.1 Work Done on Kowloon Motor Bus’s Application

By network sniffing on KMB’s app and website, some APIs were found. Figure 6 shows the interface of Web Inspector of Apple’s Safari browser.

![Figure 6. Try to sniff on the packets sent to and received from KMB server. The right pane in the Web Inspector displays the request and response information.](image)

These APIs can be accessed to obtain several types of data. All of the APIs return JSON objects, which are widely used in web applications as they are easy to manipulate, as response. Types of data available are listed below:

1. Retrieving all stops in a route, specifying the bound

![Figure 7. API for retrieving bus stops in a route with bound specified](image)
By reading the URL part in Figure 7, it can be seen that in this API, request is sent to http://search.kmb.hk/KMBWebSite/Function/FunctionRequest.ashx with parameter “action” equals “getstops”. “Route” corresponds to the requested route, “bound” means direction and “serviceType” shall be related to some special departure types.

2. Retrieving special departures information of a route
The base URL is same as API 1 in Figure 7. The parameter “action” changes from “getstops” to “getSpecialRoute”. “serviceType” is not needed.

3. Retrieving timetable of a route
Still the same base URL from Figure 7, with “action” changing to “getschedule”, together with “route” and “bound” equals the same route number and bound number.

4. Retrieving bus-bus interchange information
Same as API 3, with “action” equals “getbbiforroute”.

5. Retrieving announcements related to the route
Same as API 3, with “action” equals “getAnnounce”.

6. Retrieving ETA of a bus route at a specific stop
This is different with the above 6 APIs. The request is sent to http://etav3.kmb.hk.

Figure 8. Sniffing result of the KMB app using Wireshark, showing the API for retrieving actual ETA.
In Figure 8, it can be seen that there are 7 useful parameters sent to the server:
1. “action” = “geteta” tells server to fetch ETA
2. “lang” defines language in the response
3. “route” selects a route number
4. “bound” selects a direction
5. “stop” is some number unique to a bus stop
6. “stop_seq” is the stopping sequence number of that stop in the route
7. “serviceType” is related to special departures

Among them, APIs 1, 2, and 6 are the top 3 important APIs relating to the project. The stop list API is used to help specifying which stop we would like to get ETA. Then we can get ETA using API 6. The special departures list helps to ensure all the possible stops are in coverage. By changing parameters in the URL, different results are expected from the server when the requests are sent.

3.1.1 Testing the Results from KMB Sniffing

The abovementioned 3 important APIs in chapter 3.1 were tested and the result was positive. Desired data can now be retrieved without accessing KMB’s app or website. Using route 1A as example:

1. Retrieving all stops for Tsim Sha Tsui bound

The request was sent to http://search.kmb.hk/KMBWebSite/Function/FunctionRequest.ashx?action=getstops &route=1A&bound=1&serviceType=1. Bound number is either 1 or 2 as routes have 2 directions. ServiceType is usually 1 for normal routes and it changes for special departures. Figure 9 shows that desired results were returned.
2. Retrieving special departures

In Figure 10, a “getSpecialRoute” request was sent. In the response, “ServiceType” value was found and that would correspond to special departures. Special departures have “ServiceType” value not equal to 1.

3. Retrieving ETA of bus stop no. 7 (example) for Tsim Sha Tsui bound

For this query, we did not know what should be put in “stop” parameter. Therefore we tried to skip that parameter first:

From Figure 11, ETA data could still be retrieved without specifying “stop” parameter.

The test results have showed that the APIs are providing expected results so that they are usable in the project.
3.2 Work Done on Citybus/NWFB’s Website

The general working logic of Citybus/NWFB’s website for querying ETA was found, also by network sniffing.

![Image of Citybus/NWFB's website](image.png)

**Figure 12. The NWFB/Citybus website (left) and corresponding sniffing results (right)**

Figure 12 above shows the structure of the response sent from NWFB/Citybus’s server. It shows that the ETA data is presented in the form of a web page (which is HTML, file extension not shown in the figure). That is different from KMB which uses JSON as response format. The logic of retrieving ETA is as follow:

1. User enters desired route number
2. Server returns with route bounds and variations of routing to user
3. User picks a direction and variation of routing
4. Server returns with bus stop list of that direction and routing
5. User then picks the desired bus stop
6. ETA of that specific bus route and bus stop is shown
3.2.1 Breakdown of NWFB/Citybus’s Website

This subsection explains the APIs found in NWFB/Citybus’s website, using route A21 between Airport and Hung Hom Station as example.

1. Retrieving route bounds

A request containing desired route is sent to the server.


In the request, parameter *skey* is the route entered into the searching box. Parameter *ssid* is one of the cookies saved in user’s computer and will be used throughout the whole querying process. Parameter *sysid* is a fixed value. The server will then respond with a HTML file which shows buttons for choosing route bounds, as shown in Figure 13.

![Figure 13. Rendered response from server after searching for route.](image)

2. Retrieving variations of routing

A request asking for routing variance is sent to server.


In the request, it can be seen that *ssid* is the same with API 1 above. It is reasonably to guess that the NWFB/Citybus server uses this cookie ID to save what did the user search. *lid* specifies bounds with 0/1 for different directions. Figure 14 shows the rendered response from server.

![Figure 14. Routing variation is shown.](image)
3. Retrieving bus stops in specified routing

After selecting routing, the server returns with the list of bus stops in that routing, just like the KMB one. For the website, there are two requests doing similar function, the one with simpler response would be used.

https://mobile.nwstbus.com.hk/nwp3/showsingleroutestops2.php?info=1%7C%7CG%7C%7CA21-HUH-1%7C%7CNormal%20Routing%7C%7CCTB***A21-HUH-1***1***20***10661***I&l=1&ssid=5bff8b1254156&sysid=6

In the request, the following parameters are present:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. info</td>
<td>1</td>
</tr>
<tr>
<td>2. l</td>
<td>1</td>
</tr>
<tr>
<td>3. ssid</td>
<td>5bff8b1254156</td>
</tr>
<tr>
<td>4. sysid</td>
<td>6</td>
</tr>
</tbody>
</table>

Again, ssid is the cookie ID since API 1. The long string in info points to the desired routing. The response looks like this:

```
<iframe style="width:1px; height:1px;" onload="clearmarkers(); addstoponmap('001837',113.93707185053,22.313696602091,'S','1','1-Airport (Ground Transportation Centre)',"A21-HUH-1",'T','Y', '113.93707185053','22.313696602091'); addstoponmap('003540',113.95087454053001,22.316002122091,'0','2','2-Hong Kong Port of Hong Kong-Zhuhai-Macao Bridge',"A21-HUH-1",'T','Y', '113.95087454053001','22.316002122091'); addstoponmap('001854',114.04203738053,22.339287482091,'0','3','3-Lantau Link Toll Plaza',"A21-HUH-1",'T','Y', ... more to follow
```

In the addstoponmap() function, the number in the first parameter is the stop ID for a specific bus stop. Followed by coordinates in longitude and latitude, and some other parameters specifying stop number, stop name and which routing it belongs to. These parameters are used in retrieving the actual ETA.

4. Retrieving ETA from server

After retrieving all the required parameters, the final step is requesting for actual ETA. In this step, two requests are sent in order to get the actual time.

https://mobile.nwstbus.com.hk/nwp3/auto_af08b1e236b44c28ac962835e7e674ce.php?info=001559||A21-HUH-1||8||I&ssid=5bff8b1254156&sysid=12&_=1543473940341
This is the first request and it is for the server to log the details of the ETA query. The file name (auto_xxxxxx.php) is different every time starting a new search from scratch (i.e. accessing https://mobile.nwstbus.com.hk) and currently the naming method is not found yet. This request returns an OK message if the parameters are valid. In the info part, the stop ID, routing name, stop number are sent to the server. Again, ssid is the cookie ID used since the first step.


This is the second request. The file name is similar to the first one but not the same. Also, the naming method is not found yet and the name will change. ssid is present again and the server will send some HTML code containing the actual ETA in response, as shown in Figure 15. In the Figure, the right half is the rendered result of the HTML code sent from the server. It corresponds to the part showing ETA to users on the actual website (left half).

Figure 15. The page shown to user (left) and the actual response from server (right)
3.2.2 Testing the APIs from NWFB/Citybus Website

In the testing part, cookie ID was needed as it was required in the request parameters. For simplicity, we tried to randomly pick one previous cookie first. Figure 16 shows some cookie saved in the computer before. Try to use the third one, 5bff892e280c2, first.

![Figure 16. Picking a random previous cookie](https://mobile.nwstbus.com.hk/nwp3/routesearch.php?rtype=&skey=B5&l=1&savecookie=1&ssid=5bff892e280c2&sysid=3)

First step was retrieving route bounds. Route B5 between Hong Kong-Zhuhai-Macau Bridge Hong Kong Port and Sunny Bay MTR Station was taken as example. A request was sent to the server:

```plaintext
```

And the server returned with expected response, shown in Figure 17. Correct directions were found.

![Figure 17. Retrieving directions of route B5](https://mobile.nwstbus.com.hk/nwp3/routesearch.php?rtype=&skey=B5&l=1&savecookie=1&ssid=5bff892e280c2&sysid=3)
Then we tried to retrieve bus stop list:


Routing was found, shown in Figure 18.

![Figure 18. Routing was found for B5 HZMB direction](image)

After selecting routing, next step was to retrieve bus stops list. The info part was found in the HTML code in the response of last step. Response shown in Figure 19.


![Figure 19. Response from server after searching B5 normal routing](image)
Final step was to retrieve ETA. However, it seemed that the session has expired already and there was no result. The two requests sent:

https://mobile.nwstbus.com.hk/nwp3/auto_af08b1e236b44c28ac962835e7e674ce.php?info=003550||B5-HZM-1||2||1&ssid=5bff892e280c2&sysid=12


ETA was **NOT** shown, as in Figure 20 and 21.

![Figure 20. The response was not "OK". That indicated the query was not successful.](https://mobile.nwstbus.com.hk/nwp3/auto_af08b1e236b44c28ac962835e7e674ce.php?info=003550||B5-HZM-1||2||1&ssid=5bff892e280c2&sysid=12)

Estimated Time of Arrival is currently not available. Please refer to timetable.

**Service hours of this route please refer to timetable**

![Figure 21. ETA could not be shown. Query failed.](https://mobile.nwstbus.com.hk/nwp3/auto_0120dbe930314fdb8c81b104faa094d9.php?l=1&ssid=5bff892e280c2&sysid=14)

Therefore, it is reasonable to claim that we have succeeded in most of the steps in querying NWFB/Citybus’s ETA except the last part where the actual ETA was queried.

### 3.3 Difficulties

The KMB ETA service is relatively easy to trace the data path. There was not much hard times. However, for NWFB/Citybus’s case it was much more complicated. First, the responses from NWFB/Citybus server are HTML codes. Readability is low and it would be more difficult to extract values from the code as more attention has to be paid on formatting and character processing. The ever-changing file name of NWFB/Citybus’s API is another problem for future programming.
4 Challenges

1. After converting APK into source codes, the original naming of variables might be changed, might need to study the whole program to understand what do the variables store and their function.
2. With the limited time, the showcase application might not have a lot of functions, or the routes have to be hard-coded.

5 Conclusion

This project focuses at how to extract and make use of non-public, non-open data from smartphone applications, using ETA data from different apps from bus companies as an example. The project tries to gather the ETA data together in one place in order to benefit user in their daily lives. In this report, the general background was discussed. Then the two main techniques, Network Sniffing and Reverse Engineering were introduced. Currently, the way to gather KMB’s ETA data is found with the two important API to get stops on a route and get ETA of a specific route at a specific stop. Most of the steps in querying NWFB/Citybus’s ETA was also found, except retrieving actual ETA part failed the testing. The ultimate goal would be building a platform to integrate ETA data from the two major bus operator groups to facilitate passengers. With this project, passengers have the chance to forget about the hard times when they have to switch between applications to view bus ETA. However this project is small-scaled and it only focus on gathering and integrating data. Further speaking, we hope that more and more applications or companies will open their data to the public so that software developers can make good use of the public data and create great applications that make everyone’s life better.
6 Future Plans

6.1 Continue Work on Citybus/NWFB app and website

After finding the general logic of querying NWFB/Citybus ETA, next step is to investigate how to completely avoid actual access of https://mobile.nwstbus.com.hk from a browser in order to retrieve the cookie ID to be used in querying ETA. Also, finding a way to persist the ETA querying function so that it would not be malfunctioning after some idle time is needed. When these problems are tackled, this project can move on to next stage.

6.2 The Showcase Application

After studying and successfully found methods to obtain ETA data from both companies, the showcase application will be built. There will be two main functions to be showcased:

1. ETA of jointly operated routes
   Currently, the apps will show “it is another company’s operating cycle” (or equivalent) if the coming departures are from another company. In the showcase application, this message will not show up anymore and ETA of both companies can be shown in one single app.

2. Showing ETA of routes operated by different companies in same bus stop
   Similarly to above, for bus stops having routes operated by different companies, ETA of all routes stopping at that station will show up.

   As the network sniffing and the reverse engineering are done under Android environment, the showcase application will also be developed as an simple Android application for the ease of development. Users can enter router numbers to see ETA data.
References


