Visualizing Mobile Network Performance of Video Traffic Data

by

Chenyi Liu
B.Eng., Xiamen University, 2013

A Project Report Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

© Chenyi Liu, 2015
University of Victoria

All rights reserved. This report may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.
Visualizing Mobile Network Performance of Video Traffic Data

by

Chenyi Liu
B.Eng., Xiamen University, 2013

Supervisory Committee

Dr. Melanie Tory, Supervisor
(Department of Computer Science)

Dr. Brian Wyvill, Departmental Member
(Department of Computer Science)
Supervisory Committee

Dr. Melanie Tory, Supervisor
(Department of Computer Science)

Dr. Brian Wyvill, Departmental Member
(Department of Computer Science)

ABSTRACT

In this project, we designed an interactive web-based visualization tool that aims to support network provider’s analysts to find the underlying causes of poor network performance of video traffic data. The tool - Media Assessment Tool (MAT) Dashboard - supports customization. Users are allowed to add and remove different types of visualizations, rearrange them and apply filters to them. This project describes the visualizations, which allow users to interact with data and drill down into the data. We also describe the design process and the trade-offs of different design choices. In the end, we summarize the strengths and limitations of our design and propose possible future work directions.
Contents

Supervisory Committee ii
Abstract iii
Table of Contents iv
List of Tables vi
List of Figures vii
Acknowledgements ix
Dedication x

1 Introduction 1
  1.1 Project Overview .................................................. 1
  1.2 Problems and Requirements ...................................... 2
  1.3 Overview of The Report ........................................... 3

2 Related Work 4

3 Design Process 7

4 Prototype 9
  4.1 MAT Dashboard Overview ........................................ 9
  4.2 Visualizations ..................................................... 12
    4.2.1 Heat maps ................................................... 12
    4.2.2 Parallel Coordinates ....................................... 13
    4.2.3 Timelines .................................................... 15
    4.2.4 Correlation Matrix ......................................... 17
    4.2.5 Histograms ................................................... 18
List of Tables

Table 1.1 Use cases and corresponding tasks and targets . . . . . . . . . . 2
List of Figures

Figure 4.1 Screenshot of MAT dashboard. (a) An activated panel. (b) A deactivated and collapsed panel .................................................. 10
Figure 4.2 Hourly Heat map .......................................................... 13
Figure 4.3 Daily Heat map ............................................................. 13
Figure 4.4 Monthly Heat map ......................................................... 13
Figure 4.5 Parallel Coordinates with brushed axis ................................. 14
Figure 4.6 Timelines with small multiples. The grey brushed area in the overview determines which date range is shown in the detail views. (a) Detail view. (b) Overview. .................................................. 16
Figure 4.7 Timelines with stacking .................................................... 17
Figure 4.8 Correlation Matrix .......................................................... 18
Figure 4.9 Scatterplot Matrix with brushing and labels ............................ 19
Figure 4.10 Histograms with brushing after applying filters ....................... 20
Figure 4.11 Histograms with brushing without filters ............................. 20
Figure 4.12 The workflow of MAT Dashboard ....................................... 21
Figure 4.13 MAT Data Table Structure .............................................. 22

Figure 5.1 A visualization mockup of time series with time gap. The "accordions" on time axis indicate time gap. ................................. 25

Figure A.1 Initial state, side bar collapsed ......................................... 27
Figure A.2 Add empty panel .......................................................... 28
Figure A.3 Select KPIs for new panel ............................................... 29
Figure A.4 Six KPIs selected, choose type of visualizations. ..................... 30
Figure A.5 New chart added .......................................................... 31
Figure A.6 Users rearrange the panels .............................................. 32
Figure A.7 Users can pick a cell to see corresponding records ................... 33
Figure A.8 The corresponding details are shown in the panel as a table. .......... 34
Figure A.9 An alternative option is to show the details as a tag cloud. ........ 35
Figure A.10 Users are able to change the filters and apply the filters to each chart.
ACKNOWLEDGEMENTS

I would like to thank:

My parents, friends, and my colleagues in VisID group, for supporting me in the low moments.

Dr. Melanie Tory, for mentorship, support, encouragement, and patience.

Dr. Maria-Elena Froese and Brennen Chow, for supporting me and working together on this interesting project.

Tutela Technologies Ltd., for providing this opportunity and data.
DEDICATION

I dedicate this project report to my parents who always support me.
Chapter 1

Introduction

Global mobile data traffic reached 2.5 exabytes/month at the end of 2014, while mobile video traffic exceeded 50 percent of total mobile data and reached 55 percent by the end of 2014[1]. Network service providers (NSPs) are increasingly concerned about the performance of mobile networks and the quality of videos. Maintaining good network performance during video playback is challenging, yet it is very important to NSPs because pauses, drops in quality, or dropped frames are very noticeable and frustrating to video viewers.

1.1 Project Overview

Tutela Technologies Ltd. (Tutela) develops softwares that enable NSPs to collect and review network performance data. Tutela’s Network Assessment Tool (NAT) dashboard enables their clients to view key performance indicators (KPIs) collected from NAT in some basic charts, primarily line charts and maps. Tutela’s Media Assessment Took (MAT) supports field tests that enable the collection of approximately 40 KPIs (e.g., bit rate, dropped packets), plus relevant information of the field test (location, connection type, device, operating system, etc.). MAT is a tool specifically designed for video traffic data and it can be used to troubleshoot networking issues and support build the Content Delivery Network (CDN), eventually improving the customer experience[2]. Tutela would like to dramatically expand the analytic capabilities of the Network Assessment Tool dashboard and build a new analysis tool for their MAT that will enable their customers to explore and analyze the video traffic data.
### Use Cases

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Actions</th>
<th>Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>As an analyst, I want to select the time range and other variables (such as locations, type of devices, connection type, O.S. etc.) as filters of the charts, so I can see the network performance under different settings.</td>
<td>Present</td>
<td>Distribution, correlation, trends and outliers</td>
</tr>
<tr>
<td>As an analyst, I want to be able to select the KPIs to be plotted in the dashboard.</td>
<td>Present</td>
<td>Distribution, correlation, trends and outliers</td>
</tr>
<tr>
<td>As an analyst, I want to be able to compare the KPIs in different settings (filters).</td>
<td>Compare</td>
<td>Distribution, correlation, trends and outliers</td>
</tr>
<tr>
<td>As an analyst, I want to see troublesome networking locations.</td>
<td>Browse and explore</td>
<td>Shape and Spatial position</td>
</tr>
<tr>
<td>As an analyst, I want to see what days of the week and what time are more troublesome.</td>
<td>Summarize</td>
<td>Outliers, trends</td>
</tr>
<tr>
<td>As an analyst, I want to see what videos are more troublesome.</td>
<td>Discover and identify</td>
<td>Outliers, features</td>
</tr>
<tr>
<td>As an analyst I want to find the correlation between KPIs.</td>
<td>Identify</td>
<td>Correlations</td>
</tr>
</tbody>
</table>

| Table 1.1: Use cases and corresponding tasks and targets |

### 1.2 Problems and Requirements

Information visualization is a set of technologies that use visual representations to amplify human cognition with abstract data[3]. Visualizations can help viewers to interact and navigate the data which enable the viewers to understand the data faster and better. The end users of MAT Dashboard are the analysts of NSPs who are not experts in visualization. They are familiar with networks and the metrics of network performance (KPIs), have some knowledge of statistics, and probably have experienced other dashboards too. Basic charts like line charts and pie charts do not solve the questions they want to ask, because their primary concern is not the learnability but the effectiveness of the dashboard. Basic charts have limited channels, which is not enough for presenting complex datasets. Therefore, we need to design more effective visualization which should also not be hard for the analysts to learn.

The main use case is: As an analyst, I want to see a report with a high level overview of the data, and find out what the main networking problems are and what
caused them. Munzner proposed a framework for visualization task abstraction[4], in which actions define user goals and targets refer to some aspects of the data that interest users. Based on the main use case, after some iterations of requirements gathering, we defined the requirements and tasks described in Table 1.1. This helped us choose the suitable visual idioms and visual interactions in design process. For instance, after analyzing use case "As an analyst I want to find the correlation between KPIs", we conclude that the users would like identify the correlations of metrics, thus we should use a type of visualization that can present correlations.

The combination of geographic, time-based, multimedia, and multidimensional data plus various requirements means that a customized dashboard design is needed. The deliverables of this project include fully designed interactive mockups (user interfaces without the underlying functionality) and an interactive web prototype.

1.3 Overview of The Report

Chapter 1 contains an introduction to the background and problem this project solved, followed by an overview of the structure of the report itself.

Chapter 2 describes the related work and the limitations of current solutions.

Chapter 3 describes the design process of the project.

Chapter 4 is where the interface design and implementation is fully described.

Chapter 5 includes the discussions, contributions and limitations.

Chapter 6 draws the conclusion of the project. It also enumerates avenues of future work for further development of the concept and its applications.
Chapter 2

Related Work

The data we need to visualize in this project is not simple univariate, it is semi-structured JavaScript Object Notation (JSON) data with independent and dependent dimensions. JSON is a lightweight data-interchange format built on a collection of name/value pairs and an ordered list of values. After processing, the original data can be flattened to multidimensional multivariate (MDMV) data. We only need to visualize partial static data in this project, but scalability and dynamic data rendering should also be taken into consideration in future versions. In MDMV visualization, we can plot 2 variables in a 2D chart and expand the dimension to 3D when there are 3 variables, but experiments showed that most tasks involving abstract data do not benefit from 3D, on the contrary, 3D visualizations take more time to render and bring a certain level of distortion[4].

Many 2D techniques have been developed for visualizing multidimensional data with more than two variables, such as Scatterplot Matrix (SPLOM), Parallel Coordinates[5] and Table Lens[6]. SPLOM presents multiple adjacent scatter plots of every pair of variables. In Parallel Coordinates, variable scales are plotted on parallel axes in which a line in the table corresponds to a polyline that intersects every axis in parallel coordinates. SPLOM and Parallel Coordinates both allow users to find correlations and outliers, but the scalability is limited. The Table Lens uses Focus+Context[7] technique to help users explore large amounts of tabular data. It preserves the entire information structure(Context) when the user zooms in on specific items(Focus). These techniques all have their advantages and limitations, however we can provide users with a combination of these techniques to enable them explore and understand the data better.

Visualizing time series is another big part of the project since users need to see
patterns and changes of network performance metrics (KPIs) over a period of time. Time series are usually plotted in line charts. It is common to stack lines or areas in time series visualizations. For example, layer area graphs[8] might be used when comparing time series that share the same unit and can be summed up, and stacked Graphs[9] stack multiple time series in one view by using an algorithm. However, experimental results show that users often misinterpret the space between curves in stacked graphs[10]. A few advanced techniques were developed to minimize the misinterpretation due to overplotting. Braided Graph[11] and Horizon Graph [12] both divide and layer time series to increase the density of time series graphs. They are proved to be more effective than simply stacked graphs but still awkward for comparing tasks. Small multiples[13] is a series of similar charts using the same axes allowing them to be easily compared. It enables comparison across variables and reduces the risk of overplotting. It could take a lot of space but we can reduce unnecessary space by using interaction.

Visualization tools like Tableau are widely used for data exploration. Users can drag and drop to create visualizations in these tools, but they typically do not directly support semi-structured data like JSON data[14]. On the other hand, although generic visualization tools like Tableau allow users to do calculations and manipulate the data, the degree of freedom is limited. Tableau supports combining multiple views into interactive dashboards, but users have to start from scratch which is not easy for non-experts. Google Analytics is a web analytics service that tracks and reports website traffic[15]. It provides casual users with a high-level overview dashboard and more in-depth customization for professional users. Dashboard tools like Google Analytics enable user to start with a prefixed dashboard and then customize it, but they are not designed for generic data or network performance data.

Mobile network traffic data is increasing rapidly[1]. The metrics of mobile network performance involving video quality are quite different from wired networks, but to date, most business tools are designed for general network assessment. Little research has been done on visualizing mobile network performance of video traffic data. There are some tools on the market like Akamai[16] and Infovista[17] for analysts to monitor network performance in order to improve customer experience. Tutela also has its own tool - Network Analysis Dashboard(NAT Dashboard) - to visualize network performance metrics. NAT Dashboard is a web analysis tool which provides users with a basic time series line chart, a geographic heat map and filters to analyze network performance. These tools are effective for monitoring purpose. They provide
analysts with a dashboard of visualizations which makes it easy to find poor network performance, but they do not support further analysis including troubleshooting very well. For example, it’s hard for the network analysts to tell whether it’s the server’s problem or a problem caused by the user’s device by looking at a sudden drop of bit rate. When the dataset is large and complex, the limitation of basic static visualizations preclude showing every aspect of the data at once[4]. We need to build an interactive dashboard that enables analysts to drill down through the data and find the real causes of poor network performance. Thus, current visualization tools on the market can not solve the problem of this project. This project is designed specifically to visualize video traffic data and support troubleshooting. The specific nature of this problem and the limitations of general purpose tools meant that a new custom tool was necessary.
Chapter 3

Design Process

This project was done in collaboration with a post-doc, Dr. Maria-Elena Froese. Dr. Froese was responsible for designing the framework and visualization. I was responsible for processing the data and building the dashboard prototype. We also closely collaborated with our data provider - Tutela’s vice president Brennen Chow in this process.

Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration. It promotes continuous improvement and encourages rapidly flexible response to change[18]. SCRUM is one of the best practices in Agile software development. It is an iterative and incremental agile software development methodology focusing on maximizing the team’s ability to quickly deliver the product and respond to the emerging requirements[19]. Due to the unclear requirements of MAT Dashboard and the size of the project, we decided to incorporate some best practices of SCRUM in our design process.

A sprint (or iteration) is the basic unit of development in SCRUM. In this project, it is restricted to a specific duration of 2 weeks which means we met the company every two weeks to discuss the prototype and then refined the requirements. This project lasted for 6 months. The process was as follows:

Month 1  In the first 2 sprints, we mainly focused on gathering user requirements. We discussed with Tutela and built an understanding of analysts and use cases, then we started a literature review to find suitable visualization techniques to solve the problems. In the end of the first month, we started to design the prototype on paper.
Month 2 In the following months, we designed the dashboard for MAT and its associated visualizations iteratively using MyBalsamiq - a user interface design tool. We built interactive mockups with modifications to the paper design as needed according to the requirements, conducted low-fidelity prototyping and reviewed the mockups with Tutela employees. We regularly reviewed the design choices with Tutela through biweekly meetings in this month.

Month 3,4,5 In this phase, we developed the web prototype iteratively with modifications to the interactive mockups as necessary. Meanwhile, we refined the design choices with Tutela employees when we met some technical problems or found a better solution.

Month 6 We presented the final mockups in Appendix A and a web prototype to Tutela and refined the prototype based on their feedback, then we wrote the final report.

Our approach is iterative. During the six months of the project, we gathered feedback from Tutela staff and refined the design accordingly. This helped us save unnecessary development time and made it easier to cope with the changes. I spent a major part of the time working in Tutela’s office. When I was working there, we had daily stand-up meetings with all of the employees which made it possible to get the product owner’s feedback in time and resolve issues in advance before they become irreversible. Overall, we found that our iterative design process made it easier to deliver a satisfying quality product on schedule.
Chapter 4

Prototype

The MAT Dashboard is a web prototype designed to help network performance analysts understand the data better. We designed this fully functional prototype to provide Tutela with some design suggestions on how to visualize the data they collect. In the iterative design process, we analyzed their requirements, summarized the tasks this tool needs to support and justified the design choices with respect to alternatives.

We designed 5 visualizations in the dashboard, and each visualization supports different tasks described in Table 1.1. Heat maps support summarizing outliers and trends; correlation matrix and parallel coordinates enable users to identify correlations and outliers; time lines support discovering trends and features and histograms support discovering distributions. Users need to interact and navigate through these visualizations, and they may become lost while analyzing the visualizations. To alleviate this issue, we designed the dashboard with a fixed side bar on the left and a configurable set of visualization panels on the right. The side bar displays the corresponding settings of the current activated visualization and leaves enough space for users to navigate through all the visualizations.

4.1 MAT Dashboard Overview

The dashboard shown in Figure 4.1 has two views. View 1 on the left is a scroll-follow sidebar, users can add new visualizations and filter the data in the settings panel or select a different time range. The interactions in sidebar include:

Add a new panel
Click on the blue button ”+Add a New Chart”, there will be a new panel added in view 2. The new panel will be activated and highlighted.

**Settings**
These are all dropdown multiple selection boxes, users can select or deselect the settings by clicking on the selection items. The filters will be empty and all KPIs will be selected by default. The filters include assets - the video content, devices and operating systems.

**Time Range**
Users can select a time range from this panel as an additional filter. The two time pickers are linked which means user can not set the end time before the start time. It is set to empty by default.

**Plot By**
Users can select or change a chart type in this dropdown box, it is set to Heatmaps by default.

**Update Chart**
Clicking on this button will store the settings in side bar and update the chart.
accordingly.

View 2 is where all the visualizations are; users can explore the data by interacting with each chart and applying filters. Users can close a panel if they do not find it useful and reorder the panels by drag-and-drop according to their preference. Clicking on any part of a panel will activate and highlight it. A panel has three components:

**Panel Heading** The type of chart is displayed here, users can also click on the heading to collapse the panel if they do not want to use it temporarily. It will turn into blue if the panel is activated, for example, the Heatmaps is activated in Figure 4.1 (a) and Parallel Coordinates is collapsed and deactivated in Figure 4.1 (b).

**Panel Body** This is where the chart is plotted, users can interact with the chart. When the user interacts with a chart, the panel will be activated and the filters in the side bar will be set to the settings stored.

**Panel Footer** The filters are displayed here, so users can see it even if the panel is collapsed.

After discussing with Tutela, we agreed to visualize the top 6 most important KPIs in 5 different visualizations. These KPIs were:

**BitRate Change Requests** Count of bit rate change requests.

**Rebuffering Time** Average time spent filling the buffer while the video is not playing in milliseconds.

**Rebuffering Times** Count of when rebuffering happens.

**Delay Time** Delay time in between requesting and playing video.

**Playback error times** Count of when there is an error with the player.

**Buffer Underrun Count** Count of when a buffer is used to communicate between the device and the server.

These KPIs are the most relevant KPIs to video quality because the increase of these KPIs would cause greatest noticeable drop-off in video quality. Video quality
has a negative correlation with these KPIs, the higher the KPIs are, the worse the quality is. Thus, the visualization is designed to highlight the outliers and visualize the change in order to help users identify instances of poor network performance, and then explore the data to understand the underlying cause.

In order to explore possible causes (e.g. device, OS, time of day, etc.), we chose several variables as the filters which enables users to compare KPIs under different settings. After discussing with Tutela, we chose the following variables:

**Device**  Device model. (e.g. iPhone 4S GSM, Google Nexus 4)

**Operating System**  Operating system name and version. (e.g. iOS 7.0.1, Android 4.4.3)

**Asset**  The name of the video source.

**Timestamp**  The time when the user started to play a video.

Users have control over all the filters including 6 KPIs, time range, device type, operating system and video content. The final product would allow access to all KPIs and other possible setting options including geographic location and connection types (Wifi/3G/4G).

### 4.2 Visualizations

#### 4.2.1 Heat maps

The heat maps is the first default visualization on the dashboard, it shows the trend and patterns of video quality by different time granularity - hourly (Figure 4.2), daily (Figure 4.3) and monthly (Figure 4.4). Users can switch between these time granularity by clicking the radio buttons in the panel. This visualization supports tasks including finding outliers and summarizing data.

We calculated the average KPIs of a certain time range and then colored the corresponding cell, hovering the mouse on the cell shows the exact number. Heat maps are a compact visualization idiom that use matrix alignment; each cell is colored by the corresponding value. In North American culture, red is seen as bad or warning, so the color scheme in this heat map is from yellow to red - good quality to poor quality. For example, in Figure 4.3, we can tell the network quality is relatively worse
on the weekend. Since this is the first visualization users will see when they open the dashboard, it should provide overview of the dataset. Heat maps encode quantitative data with colors and do not take much space, which is good for providing a high information density overview.

### 4.2.2 Parallel Coordinates

Parallel Coordinates plot each row in the data table as a polyline; each attribute of a row is plotted on the corresponding vertical axis as a vertex of the line. Attribute scales are plotted on parallel axes. The tasks parallel coordinates support include finding trends, outliers, extremes and correlation. Linking and brushings refers to the connecting of two or more views of the same data[20]. Brushing means selecting
Figure 4.5: Parallel Coordinates with brushed axis
a subset of the data items to highlight this subset, linking causes the brush effect like highlighting to be applied on the same data items in other parts of the visualization. For instance, users can brush on the axis in parallel coordinates to select partial data, then the selected lines will be highlighted and others will be greyed out as shown in Figure 4.5. It gives users an overview of all variables as well as the range of an individual variables. Users can also identify outliers and correlations in Parallel Coordinates, but we used a correlation matrix which will be discussed in section 4.4.4 to further support these tasks.

Parallel Coordinates is able to present hundreds of records at once, but the width of the screen is fixed, so the number of variables in a parallel coordinates is limited. CDNs deliver web content including videos to a user based on the geographic locations of the use. According to Tutela’s requirements, assets (videos) play an important role in CDN planning. In order to meet this requirement, we plotted this important discrete variable on the last axis - assets. At first, we used colors to encode assets and put a legend next to the parallel coordinates, but it’s hard to find the lines for assets that did not occur often. Brushing means selecting partial data from a visualization. For example, in Figure 4.5, the lines related to video ”Sintel” are covered by lines related to video ”Nasa” without brushing. Therefore, the last axis on the right also supports brushing, so users will be able to select one or more assets and see network performance of certain assets without applying filters.

4.2.3 Timelines

Since users need to see KPIs change along the time line, we designed this visualization to support this task. In this chart, we used small multiple line chart to display each KPI’s time series data and the lines are colored according to their status in the detail view (Figure 4.6 (a)). We also added a heat map overview (Figure 4.6 (b)) at the bottom of small multiples. When users choose less KPIs to visualize, the height of each details view would expand to fill in the space accordingly. We could have stacked all the lines in a large details view but it became messy and less effective as shown in Figure 4.7. Instead of using a simple thumbnail image as the overview, we chose to use a slender heatmap metaphor as the overview in order to save space and avoid visual clutter. This idea was inspired by Dr. Heidi Lam’s Line Graph Explorer[21], in which she used Focus+Context technique. Due to the space limitation in our dashboard, we decided to use Overview+Detail to enable users do more in-depth exploration.
Figure 4.6: Timelines with small multiples. The grey brushed area in the overview determines which date range is shown in the detail views. (a)Detail view. (b)Overview.
Figure 4.6 shows that when user brush on the overview, the time axis in detail view will change along with it. Therefore, if a user finds a problematic time segment in the overview, he or she can brush on that area to see more detailed KPI change in that time range.

4.2.4 Correlation Matrix

Correlation is a statistical measure of the relationship strength between two datasets. So if we have one dataset $X = \{x_1, \ldots, x_n\}$ containing $n$ values and another dataset $Y = \{y_1, \ldots, y_n\}$ containing $n$ values then that formula for $r$ is

$$r = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n \sum_{i=1}^{n} x_i^2 - \left( \sum_{i=1}^{n} x_i \right)^2} \sqrt{n \sum_{i=1}^{n} y_i^2 - \left( \sum_{i=1}^{n} y_i \right)^2}}$$

(4.1)

where $n$ = the number of data pairs, $x_i$ = a number in dataset $X$, and $y_i$ = the corresponding number in dataset $Y$.

In MAT dashboard, we use it to find the correlation between two KPIs. Correlation between two KPIs can be calculated by Pearsons Correlation Coefficient formula or visualized by a scatter plot.

The correlation value $r$ ranges from $-1$ to $1$, $-1$ is considered to be perfectly negative correlation, $1$ is considered to be positive correlation, and $0$ means these
two groups of numbers have no linear relationship. In correlation matrix, we color each cell by its corresponding correlation value from blue(−1) to red(1). Clicking on the cell populates the corresponding scatter plot; the dots in the scatter plot are colored by assets. Figure 4.8 shows the default status of the correlation matrix, it prepopulates the scatter plot with the highest correlation value.

The Scatterplot Matrix (SPLOM) is considered to visualize the correlation between KPIs. Given k variables, the SPLOM is a k*k matrix, each row and column defines a single scatter plot. Linking and brushing in a SPLOM can be very useful. It means that when the dots in one scatter plot are highlighted, the corresponding data (the variables of the same row) in other scatter plots will be highlighted as well. Other dots will be greyed out. Figure 4.9 shows an example of linking and brushing in a scatter plot. Although it can be used for identifying correlations and outliers, it takes too much space and as the number of variables increases, the dots and scales become hard to identify. The rendering time also increases.

4.2.5 Histograms

Histograms display the distribution of each KPI. Linking and brushing would allow the user to select and highlight one bar of a histogram, then corresponding portions of bars are also highlighted in the other histograms. Figure 4.10 is an example of brushed histogram. Research has shown that brushing histograms are effective for complex exploration tasks.[22] When it comes to the MAT data, brushing histograms were not as effective as we thought. The network performance is good in most of the cases, so without applying filters, there is always a high bar in histograms which indicates good quality (Figure 4.11). The end user’s goal is troubleshooting, so visualizations should
Figure 4.9: Scatterplot Matrix with brushing and labels
be designed to help them focus on KPIs related to poor network quality. The high bars in the default histograms draw too much attention, but did not provide useful information. The scale of the histograms makes the smaller bars almost invisible which is what the analysts want to focus on, so we decided not to use histograms in default visualizations. However, they could become more useful after applying filters if the chosen settings happen to be the potential causes of unstable network performance.

4.3 Implementation Details

This web application - MAT Dashboard - is hosted on a local server. The data in JSON format can be fetched by a given link and stored in local host. The client side web application is written in HTML, CSS 3 and JavaScript. The framework is based on Bootstrap 3 - a framework for developing responsive, mobile first projects on the web[23]. The visualization library we used is D3.js[24]. We chose these techniques because they are open source, very accessible and flexible. D3.js supports advanced interactions, and there are also a lot of Bootstrap plug-ins to support framework infrastructure.

Tutela’s Media Assessment Tool (MAT) is a mobile application to monitor video performance. First, the network provider crowdsources network performance information from the perspective of actual customers. Then, customers can use the application and report performance data to the server, so the network providers’ analysts will be able to fetch the data from the server and analyze the data using MAT dashboard.
Figure 4.12: The workflow of MAT Dashboard

The process is shown in Figure 4.12.

The data collected and recorded by the MAT is stored in a database on the mobile device. This database contains several tables that are linked together. Once the data is generated, existing data will be packaged as a JSON object and sent to the server via an HTTP POST Request. In this report, we call this JSON object an event. MAT Dashboard retrieves all the events and downloads them as an array of JSON objects. There are several linked tables in the events, but not all of them can be linked together, so we only visualized a portion of the data as illustrated in Figure 4.13 [25].

MAT Dashboard flattens the tree-like linked table and uses some of the attributes including network performance metrics (KPIs) like Bitrate Change Request, Rebuffering Time, ID, time stamp and other derived data as new fields of the flattened table. Every event is a tree like object and it is derived to several items in the flattened table according to the number of leaf nodes it has. This process will happen when the user opens the dashboard, the flattened table will be stored in memory as a global variable. Then, we bind the processed data to the default visualizations so that users will be able to see a few default visualizations once they open the dashboard.
Figure 4.13: MAT Data Table Structure
Chapter 5

Discussion

As a visualization tool that was designed specifically to meet Tutela’s potential customers needs, initial feedback from Tutela indicates that our prototype and design provides an intuitive way to do network performance analysis. Compared to the visualization tool they developed before, this tool is expected to be more effective and user friendly. In general, Tutela were satisfied with the visualization tool. During the design process, we got feedback from Tutela frequently; we met with Tutela’s vice president Brennen Chow bi-weekly and I also got regular feedback from several other Tutela employees. It is critical to get real users involved in the design process, unfortunately, we did not have the chance to talk to Tutela’s potential customers - NSPs - for business reasons. So in the future, this prototype may not meet the end users’ requirements perfectly, but it provides some design suggestions.

The main purpose of this tool is to provide users with multiple visualizations to assist analysis. All the visualizations should be resizable and can be rearranged on the web page, but it becomes difficult to design the default layout. At first, we fit all the charts on the screen but they turned out to be blurry on smaller monitors. The labels became hard to recognize because the information density in these visualizations are relatively higher than conventional charts like pie charts. Then we fit them in a vertical linear layout. In this layout, every chart was clear and easier to interact with, but it became hard to fit more than two charts on the screen. Thus, the linear layout made it harder to compare different visualizations. There is a trade off between presenting all possible aspects of data and presenting a detailed aspect of data. Overall, this problem can be solved by making the charts interactive, but we chose the linear layout for this prototype to improve the readability of visualizations. Tutela will be able to optimize the layout in future versions.
Scalability is another issue we encountered in the design process. The current size of the database sample we have is 20.1 MB, in which there are 3320 test events, each event contains several sessions. After processing, the original dataset was derived to a table with 13 attributes and 2414 instances. The derived data takes 2.5 seconds on average to finish loading the default visualizations in Chrome 43.0. For only 20 MB data, 2.5 seconds seems relatively long. Nowadays, data mining and machine learning are widely used to process large amounts of data. However, it is not easy for business users to use these techniques or understand the results. Data visualization is complementary to data mining which allows a faster data exploration and learning process. Visualization is more effective when users don’t know what sort of pattern they are looking for. The dashboard we designed currently supports 6 KPIs and is able to support tens of KPIs at most. The visualizations become blurry and hard to recognize if the number of KPIs exceed this limit. We also abandoned the Scatterplot Matrix due to limited screen space. The Scatterplot Matrix was first considered to display correlations but later we found it took too much space and the brush took too long to render. Correlation Matrix and Parallel Coordinates together were proposed to replace the Scatterplot Matrix. In Parallel Coordinates, each axis can have at most two neighboring axes which limits the pairwise relationships between variables that can be explored at once. Users must reorder the axes to explore more complex relationships. In the Correlation Matrix, every possible pair of relationships are displayed at once. Users are able to click on the cell to view the two dimensions in a scatter plot. Thus, a combination of these two techniques gives an overview of all correlations while also enabling users to see all possible correlations in detail.

When we were designing timelines, a less investigated problem was discovered - sparse and discrete time series visualization. In the sample data we have, there are some huge time gaps between test events and there were a lot of test events over a short period of time. As the result, line charts become bumpy and the interpolation between two discrete points (time gap over a month) became misleading. Therefore, we designed a few alternative choices. First, we tried to use step interpolation for time series that alternating between vertical and horizontal segments, as in a step function. The step interpolation suggests the KPIs remain the same during the time gap whereas there were no test events happening during that time. Then we came up with two strategies, both of which visualize every test event separately rather than taking an average value. One choice is to use "accordions" shown in Figure 5.1 present time gap; the dot line interpolation between the corresponding points suggest
there is no data in that period of time and the interpolation is invalid. The other choice is to visualize every event in a scatter plot using position and size of the dot to convey information. Both of these two strategies are relatively new and we could not find enough experimental results to prove they are more effective than the one we chose in the final prototype.

Figure 5.1: A visualization mockup of time series with time gap. The "accordions" on time axis indicate time gap.

Tutela was not concerned about this issue because the gap between test events will not be long in a full dataset, so we will have a smooth line. However, this inspired us to explore an interesting question - how to visualize sparse time series? We strongly suspect that these gaps would continue to exist in a more complete dataset, even if less frequently. Moreover, they occur in many other types of time series data as well. For example, earthquakes and hurricanes happen occasionally not continuously. There are some earthquake visualizations projects[26] which visualize each event by time but most of them visualize events in an aggregated static view[27]. There has not been much work done to visualize sparse time series data in an interactive way, so this would be an interesting direction for future work.

There are some changes I recommend that Tutela make before deploying this project to their customers. First, data processing should be done in the back-end or on the server side. In this project, we processed all the data in the front-end which is time consuming due to the limited computational capabilities of browsers. If this process have been done on the server, it would save the end user a lot of time. Second, front-end developers should start using real data as soon as possible. Using fake data for visualizations in rapid prototyping and iterative design process is effective at the beginning of the project. However, developers should use the real data as soon as possible, because some visualizations might become surprisingly ineffective after using the real data. Get the real data involved earlier in the design process also can help estimate the scalability of the system and make adjustments sooner.
Chapter 6

Conclusions and Future Work

The novel visualization tool (MAT dashboard) described in this report integrates multiple commonly used visualization techniques and more advanced techniques to support analysis of mobile network performance for video traffic. In the iterative design process, we tailored our visualization tool corresponding to changing requirements.

In the future, additional features (e.g. resizing the panels, displaying events table in visualization, linking and brushing some visualizations) can be added to the dashboard to make it more user friendly. Exploring more design options and visualizations could be useful too. Using data mining or machine learning in conjunction with visualizations in the future could be more insightful when dealing with large amounts of data. Future work also includes conducting a user study or usability test with potential end users. The results will be helpful to improve functionality and user experience in order to better support network performance analysis.

We designed a compact Overview+Detail visualization for time series data. We found a less investigated visualization problem during the design process - how to visualize sparse time series. This is an interesting topic worth investigating in the future because discrete sparse time series data is common in many applications.

This combination of perspectives (time, space, and multidimensional data) is not unique only to network performance data. Thus, our design ideas may be applicable to other problems in the future.
Appendix A

Appendix: The list of mockups

These mockups are designed by Dr. Maria-Elena Froese. In the web interactive prototype, clicking on the blue area in a mockup leads to the next mockup.

Figure A.1: Initial state, side bar collapsed
Figure A.2: Add empty panel
Figure A.3: Select KPIs for new panel
Figure A.4: Six KPIs selected, choose type of visualizations.
Figure A.5: New chart added
Figure A.6: Users rearrange the panels
Figure A.7: Users can pick a cell to see corresponding records.
Figure A.8: The corresponding details are shown in the panel as a table.
Figure A.9: An alternative option is to show the details as a tag cloud.
Figure A.10: Users are able to change the filters and apply the filters to each chart
Bibliography


Glossary

**brushing** is the interaction of selecting data items from a visualization. vii, 18, 21

**CDN** Content Delivery Network (CDN) is some distributed servers that deliver web pages and other web content to a user based on the geographic locations. 1, 15

**Focus+Context** The basic idea with Focus+Context visualizations is to enable viewers to see the object of primary interest presented in full detail while at the same time getting a overview impression of all the surrounding information - or context - available. 4, 18

**heat map** A heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors. 5, 14

**HTTP POST Request** method is designed to request that a web server accepts the data enclosed in the request message’s body for storage. 10

**JSON** JavaScript Object Notation. 4, 5, 10

**KPI** key performance indicator. 1–3, 5, 10–12, 14, 18, 19, 21, 24

**linking and brushing** The idea of linking and brushing is to use interactive changes made in one visualization are automatically reflected in the other visualizations. 15, 21, 26

**MAT** Media Assessment Tool (MAT) is a tool supports field tests and collects network performance data of video traffic. 1, 10, 26

**MDMV** multidimensional multivariate. 4
**NAT** Network Assessment Tool (NAT) is a tool that collects network performance data. 1, 5

**NSP** network service provider. 1

**Overview+Detail** Two images are used for presentation. One shows a rough overview of the complete information space and neglects details. The other one shows a small portion of the information space and visualizes details. Both images are either shown sequentially or in parallel. 18, 26

**SCRUM** is an iterative and incremental agile software development methodology for managing product development. 7

**SPLOM** Scatterplot Matrix. 4, 21

**Tutela** Tutela Technologies Ltd.. 1, 10, 15, 23, 25