The Report Committee for Cheng Qian
Certifies that this is the approved version of the following report:

Classification of Encrypted Cloud Computing Service Traffic Using Data Mining Techniques

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Joydeep Ghosh

Kasi Narayanaswamy

Classification of Encrypted Cloud Computing Service Traffic Using Data Mining Techniques

by

Cheng Qian, B.E.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2011

Abstract

Classification of Encrypted Cloud Computing Service Traffic Using Data Mining Techniques

Cheng Qian, M.S.E.

The University of Texas at Austin, 2011

Supervisor: Joydeep Ghosh

In addition to the wireless network providers' need for traffic classification, the need is more and more common in the Cloud Computing environment. A data center hosting Cloud Computing services needs to apply priority policies and Service Level Agreement (SLA) rules at the edge of its network. Overwhelming requirements about user privacy protection and the trend of IPv6 adoption will contribute to the significant growth of encrypted Cloud Computing traffic. This report presents experiments focusing on application of data mining based Internet traffic classification methods to classify encrypted Cloud Computing service traffic. By combining TCP session level attributes, client and host connection patterns and Cloud Computing service Message Exchange Patterns (MEP), the best method identified in this report yields 89% overall accuracy.

Table of Contents

Chapter 1: Introduction

1.1 Cloud Computing Overview

1.1.1 Definition

National Institute of Standards and Technology (NIST) acknowledges that "Cloud Computing is still an evolving paradigm" and provides the following definition attempting to encompass all of the various cloud approaches: Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

The NIST definition also describes essential characteristics and service models of Cloud Computing. The content in this report is more related to the service model of Cloud Software as a Service (SaaS) - an application are not physically hosted in the users environment but are managed in the Cloud. Furthermore, a Cloud Computing application may not function by itself, but depend on other applications in the Cloud. The collaboration among Cloud Computing applications makes it possible to create more comprehensive applications that meet the challenges with ever increasing complexities from the real world. The collaboration also encourages reuse of existing capabilities in the Cloud, thus reduces the Time-to-Market (TTM) when building a new application and lowers the cost.

1.1.2 Web Service

The close collaboration among the Cloud Computing applications requires an integration framework that can exchange messages in a standardized way. The

standardized integration framework should cover the aspects of describing messages, exchanging messages and processing them. The integration framework should also be technology natural, therefore allow all participants using different technologies, e.g., programming languages, network protocols and development tools. Lastly, the integration framework should also include a security feature to protect user privacy and sensitive business information.

Web Service technology standardized by Word Wide Web Consortium (W3C) fits the need of the above mentioned integration framework for the Cloud Computing. W3C defines Web Service as the following: A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format, specifically Web Service Definition Language (WSDL). Other systems interact with the Web service in a manner prescribed by its description using Simple Object Access Protocol (SOAP) messages, typically conveyed using Hypertext Transfer Protocol (HTTP) with an Extensible Markup Language (XML) serialization in conjunction with other Web-related standards [2].

Web Service gained its popularity very quickly in the enterprise information system domain and serves as the standard for B2B data exchange. But the requirements imposed by SOAP based web service, e.g., XML message handling, are deemed as too heavyweight and counterintuitive by many web application developers. Roy Thomas Fielding's, doctoral dissertation [3], *Architectural Styles and the Design of Network-based Software Architecture*, describes Representational State Transfer (REST) as a key architectural principle of the World Wide Web, and has received a large amount of attention. A RESTful web service (also called a RESTful web API) is a simple web service implemented using HTTP and the principles of REST. A RESTful web service explained by Lenoard Richardson and Sam Ruby in the book titled with *RESTful Web*

Services [4] has the following aspects: (i) the base URI for the web service, such as http://foo.com/resources/; (ii) the Internet media type of the data supported by the web service. This is often JavaScript Object Notation (JSON) or XML but can be any other annotations; (iii) and the set of operations supported by the web service using HTTP methods (e.g., GET, PUT, POST, or DELETE).

RESTful service now is so popular that it exceeds the SOAP based web service according to the open APIs exposed by main Internet application providers, e.g., Amazon, Twitter and Google.

1.1.3 Encryption in Cloud Computing

Security feature is a main factor of concern when developers build Internet applications nowadays. ITU-T Recommendation X.800, Security Architecture for OSI [5], defines a systematic approach of assessing security needs of an organization and evaluating and choosing various security productions and policies. X.800 illustrates the concept of passive attack and active attack: a passive attack described by X.800 attempts to learn or make use of information from the system but does not affect system resources; while active attack attempts to alter system resources of affected their operation.

The Transport Layer Security (TLS) protocol [6] has been widely adopted by the Cloud Computing to address both passive attack and active attacks. The TLS protocol provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. The TLS protocol itself is based on the SSL 3.0 Protocol Specification as published by Netscape. IETF RFC 2818 [7] describes HTTPS that uses TLS to secure HTTP connections over the Internet.

Because HTTP is the main communication protocol for the Web Service (both SOAP based and REST based), HTTPS becomes the main technology to secure the Cloud Computing traffic. In today's word, most user privacy (e.g. identity, location, payment and etc.) and confidential business data are encrypted by HTTS/TLS when they are transferred among Cloud Computing applications.

1.2 Traffic Classification

1.2.1 Definition

Internet traffic classification is to associate the observed traffic with a specific application, and the classification results are used for profiling network usage and controlling the traffic under institutional policies etc. [8]

1.2.2 Motivation for Encrypted Cloud Computing Traffic Classification

The concept of "net neutrality" means that Internet Service Providers (ISPs) should treat all sources of data equally. It has been the center of a debate over whether those companies can give preferential treatment to content providers. In the most recent FCC rule, FCC give room to wireless operators to control the traffic, while ban any outright blocking and any "unreasonable discrimination" of Web sites or applications by fixed-line broadband providers. Wireless Internet Service Providers need the traffic classification before they apply any treatment to the traffic.

In addition to the wireless network providers' need for traffic classification, the need is also more and more common in the corporate market. Most companies today have distributed applications that are deployed to different office locations or data centers. Corporate needs to do traffic classification to shape and prioritize the traffic based on the importance of the application.

The need of traffic classification is very important in the Cloud Computing environment. Applications hosted in one data center (e.g. Amazon Elastic Cloud) have different priority and Service Level Agreement (SLA) agreed between the clients and Web Service providers. Data centers need to have the capability of traffic classification so they can guarantee the service quality provided to premium users and achieve overall network efficiency.

More and more Cloud Computing applications deal with user privacy data and sensitive business data, thus application of encryption in Cloud Computing service traffic is overwhelming. Another driving factor is the impending adoption of IPv6. Today, Internet is based on IPv4 network and it's running of IP addresses. The most advertised feature of IPv6 is the larger address space to solve the IP address shortage issue. IPV6 also provides enhanced security features. IETF IPv6 specification [12] mandates a full implementation of IPv6 to include implementation of the IPSec [13] security headers. IPSec will encrypt the whole messages being transferred between network nodes. Therefore, the ability of classifying encrypted Cloud Computing service traffic will be one of the main requirements of today and future's Cloud Computing traffic classification.

1.2.3 Related Work

The survey of techniques for Internet traffic classification using machine Learning conducted by Nguyen et al. [9] gives a broad overview of how research community responded to the traffic classification problem by investigating classification schemes capable of interring application-level usage patterns without deep inspection of packet payloads. It surveys significant works in the field of machine learning based IP traffic classification during the peak period of 2004 to early 2007. It shows a promising result of

machine learning based IP traffic classification that is applicable to both offline and online IP traffic classification. For the purpose of this report, we're more interested in the methods that can be applied to real time classification case where immediate decision needs to be made, e.g. the work done by Barnaille et al [17] and the work done by Nguyen and Armitage [18].

The paper *Internet Traffic Classification Demystified: On the Sources of the Discriminative Power* authored by Lim et al. [10] studies similar machine learning based approaches as studied by Nguyen and they reveal the three sources of the discriminative power in classifying the Internet application traffic: (i) ports, (ii) the sizes of the first one-two (for UDP flows) or four-five (for TCP flows) packets, and (iii) discretization of those features. The port will not be useful in this report because 443 (HTTPS) are widely used by most of the encrypted Cloud Computing web service traffic. However, the other two factors will be useful for this report.

Lim's paper states that C4.5 (Decision Tree algorithm) performs the best under any circumstances, as well as the reason why: because the algorithm discretizes input features during classification operations. It also pointed out that he entropy-based Minimum Description Length discretization on ports and packet size features substantially improve the classification accuracy of every machine-learning algorithm tested. In the experiments of this report, the C4.5 decision tree algorithm is adopted as the algorithm for Cloud Computing traffic classification.

Gu and Zhang's Paper titled with *Encrypted Internet Traffic Classification Method based on Host Behavior* [11] pointed out the challenges of classifying general encrypted traffic: (i) different flow in the same application may have different flow statistics due to application complexity, (ii) some flows in a given application do not have obvious and specific flow statistics. Those challenges cannot be addressed by traditional

classification methods; thus, Gu and Zhang proposed a new approach based on analyzing host behavior. It takes two aspects to improve the accuracy and speed of this method for network traffic classification: (i) observing statistics of individual flows and build IP flow profile for a given application, which describe the communication patterns of this application. (ii) use source-destination IP pairs and connection characteristics to classify the traffic with high accuracy and faster computational time.

The aforementioned works support general Internet traffic classification while this report focuses on a specific domain - encrypted Cloud Computing service traffic classification. Features that are specific to encrypted Cloud Computing service traffic are employed to improve the performance of traffic classification.

1.2.4 Main Challenges

In the problem domain of classifying encrypted Cloud Computing service traffic, the two challenges illustrated by Gu and Zhang [11] are also applicable. A typical Google map service contains many operations and the traffic related to each operation shows different flow statistics; and even for a given web service operation, it may accept different input parameters and the response messages corresponding to the input parameter may have different length. For example, a user may ask for a satellite map than a normal map in a map search request and the satellite map response results in a much bigger flow statistics than a normal map.

In addition, a host (identified by an IP) may provide more than one kind of services. E.g., Google provides search service, map service, album service, social networking service and many other types of web services. Many of those services could be exposed through the same Internet address. Thus, it's very difficult to simply categorize the services based on IP address. Furthermore, the services provided by the

same service provider may share some common operations so that different web services may be misclassified as one service. For example, Google and Twitter heavily use OAuth, an open protocol to allow secure API authorization in a simple and standard method from desktop and web applications, as the protocol to authenticate the web service consumer.

1.3 Scope of The Report

The experiments presented in this report have the following scope:

- Focus on studying the Web Service based Cloud Computing service traffic
- Focus on studying the TLS encrypted Cloud Computing traffic
- Focus on TCP traffic.
- Compare the classification result by applying existing Internet traffic classification techniques
- Identify Cloud Computing specific characteristics that can be used to optimize existing Internet traffic classification methods.

The experiments do not include any analysis of encrypted IPv6 traffic.

1.4 Content of This Report

Chapter 2 of this report describes the heuristics used for analyzing the captured traffic. It starts with explaining existing mainstream Internet traffic classification methods and how to adapt them to solve the problem discussed in this report. Then, it describes specific Cloud Computing characteristics that can be used to optimize the classification methods. The last part of the chapter describes the methodologies for comparing the results from different mining methods. Chapter 3 depicts the test environment and explains how the traffic data is captured. One of the main challenges of analyzing encrypted data is to understand the ground truth because a traffic-capturing node has no

visibility to the payload, i.e., it does not know to which Cloud Computing service a captured packet belongs. Therefore, Chapter 3 covers how to set up the ground truth. Chapter 4 presents test results and compares the output of different mining methods. It also shows different traffic patterns observed during the testing. Chapter 5 concludes this report by providing a summary and discussing future work.

Chapter 2: Heuristics and Methodologies

2.1 Heuristics

2.1.1 Using Fast Decision Tree C4.5 Algorithm

Fast Decision Tree C4.5, SPRINT algorithm [16], is the recommended algorithm per the discussion in the related work chapter and it will be used throughout the experiments. The usage of a single algorithm also attributes to a common baseline shared by all different methods.

2.1.2 Using TCP Header Attributes

We start with using TCP header attributes (flow features, source IP address, source port, protocol, destination IP address, and destination port) as input to the classification algorithm. The result of this mining method serves as the baseline so later we can evaluate how much the classification performance has been improved by using different mining methods.

We foresee that source IP address and source port will not contribute to the classification in a client-server architecture dominant Cloud Computing world. The source IP addresses may come from anywhere of the Internet for a popular Cloud Computing service and the source ports are randomly assigned by the client Operating Systems. Destination IP address will be helpful for identifying the provider of a Cloud Computing service (e.g. Google or Twitter), but it does not contribute much to differentiate services from the same provider. The destination port number will be useless since most HTTPS/TLS traffic use port 443. In a nutshell, all the parameters in this method will be of low value to the Cloud Traffic classification.

2.1.3 Using IP Flow Statistics

Using statistics of the IP flow (data packet size of first N number of packets). In the experiments, we will try different N number in order to identify the number that strikes the balance between classification correctness and processing overhead. We will also consider how the application of TLS will impact the number N selection. TLS traffic type is recognizable from the traffic-capturing node and the TLS handshake messages (negotiating encryption algorithms and keys between client and server) always precede the encrypted application data. It's a fact that the number of packets related to TLS handshake varies depending on the TLS implementation and whether the TLS session is a newly created TLS session or a resumed TLS session. The experiments need to consider the following two scenarios: (a) using IP flow statistics related to TLS handshake and application data packet; (b) using IP flow statistics related to application data packet only.

2.1.4 Analyzing Host Behavior - IP flow profile

Analyzing host behavior includes observing statistics of individual flows and building IP flow profile for a given application. Figure 1 below illustrates the entities related to a typical IP flow of encrypted Cloud Computing.

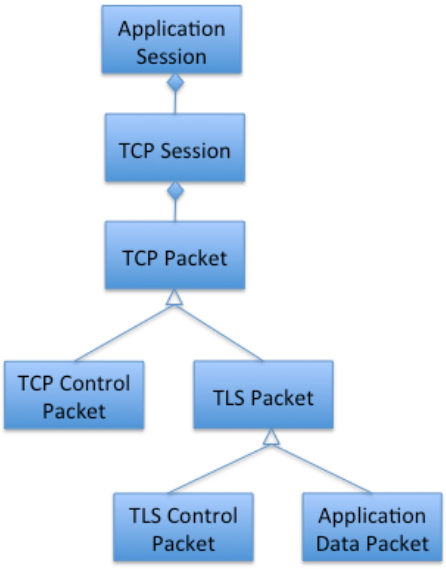

Figure 1: Encrypted Cloud Computing Traffic Flow Profile

Application session sits at the highest level and it denotes a web service invocation. A typical web service invocation may include a pair of request and response messages exchanged between client and server, but it may also have other message exchange forms. The message exchange patterns of an application session will be further described in later heuristics. Application session cannot be determined by a traffic-capturing node and it's only visible to Cloud Computing web service clients and servers. An Application Session can be related to one or more TCP sessions.

A TCP session is uniquely identified by a tuple comprised of source IP address, source port, protocol, destination IP address and destination port. A traffic-capturing node can recognize a TCP session. Within a TCP session, there are one or more TCP packets.

A TCP packet has the attributes of packet length (in bytes) and direction: client to server or server to client. A TCP packet may have two forms: TCP control packet, e.g.

TCP ACK, FIN and etc. and TLS traffic, which is the payload transferred by TCP protocol. Traffic-capturing node has the capability of knowing TLS traffic type: TLS control packet (e.g. used for TLS handshake) or Application data, which is the payload protected by TLS protocol.

2.1.5 Analyzing Host Behavior – connection pattern

Another perspective of host behavior analysis is to observe source-destination IP pairs and connection pattern. Figure 2 below shows connection patterns occurred in the experiments of this report.

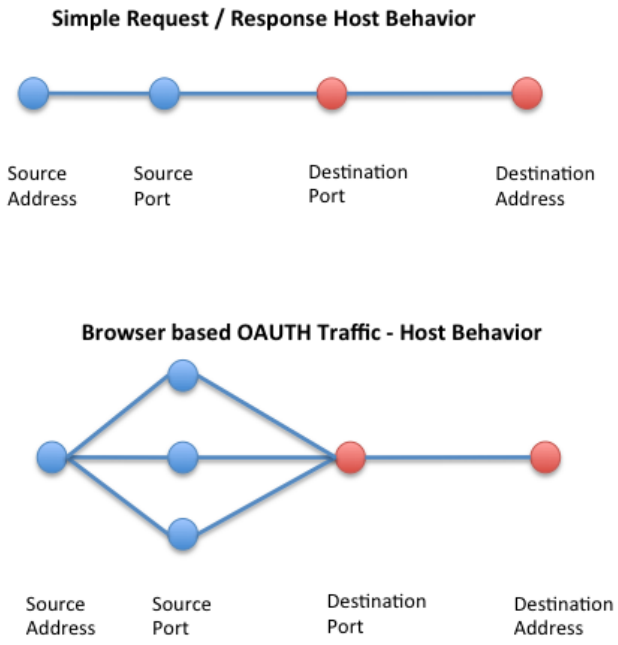

Figure 2: Encrypted Cloud Computing Traffic TCP Connection Pattern

The simplest form of TCP connection for a web service request / request pair that is one source address/source port is associated with one destination address / destination

port. While a web browser based OAuth implementation may have multiple source ports corresponding to one source address / source port pair.

2.1.6 Cloud Computing Service Characteristics

Considering Cloud Computing specific characteristics can optimize the performance of existing traffic classification methods. Thomas Erl summarizes web service Message Exchange Patterns (MEP) in the book named *Service-Oriented Architecture Concepts, Technology, and Design* [14]. A primitive MEP can have the form of request-response or fire-and-forget. The destination of a single web service request can be a single destination, multi-cast or broadcast. And a complex MEP is comprised of more than one primitive MEP. The most common complex MEP is publish-and-subscribe model, which is a combination of request-response MEP and Fire-and-forget MEP. The message exchange patterns are concepts that belong to Application Session level depicted in Figure 1 above.

Figure 3 below illustrates the Message Exchange Pattern we plan to test in the experiments of this report. The testing covers primitive MEP fire-and-forget and request-and-response and the complex publish-and-subscribe MEP. A fire-and-forget message exchange results only unidirectional TLS application data packets. Please note that the TLS control packets are still bidirectional in this case. Request-and-response message exchange results bidirectional TLS application data packets. Subscriber-and-publish message exchange firstly triggers bidirectional TLS application data traffic similar to request-and-response MEP and later unidirectional TLS application data (from server to client), which is similar to fire-and-forget message exchange.

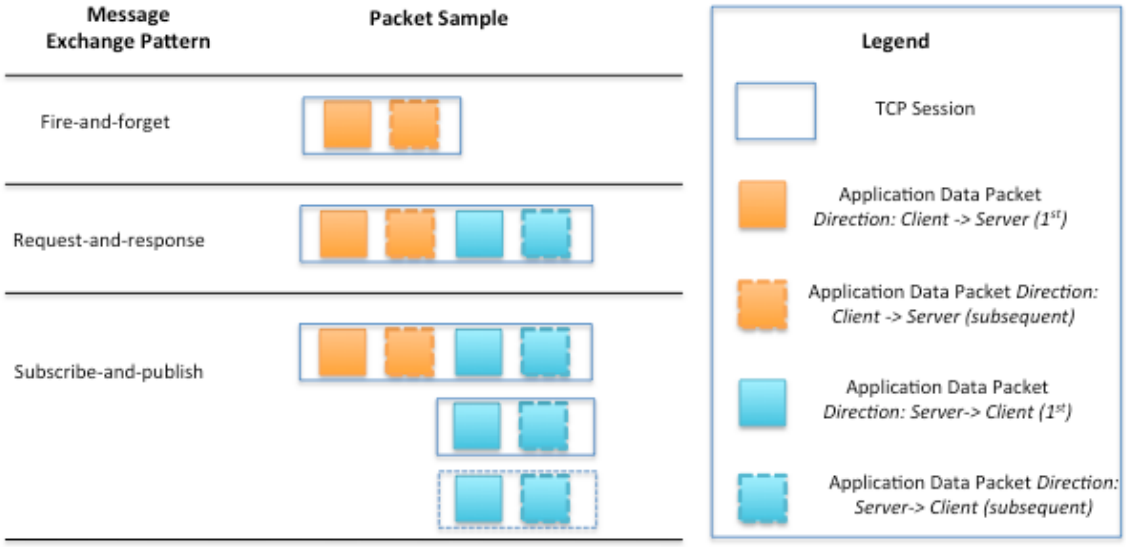

Figure 3: Message Exchange Pattern (MEP)

2.2 Methodologies for Evaluating Test Results

Lim et al. defined a set of metrics for evaluating traffic classification performance of machine learning algorithms: overall accuracy, precision, recall, F-measure, and classification speed:

(i) Overall accuracy: the ratio of the number of correctly classified traffic flows to the total number of all flows in a given trace. This metric is to measure the accuracy of a classifier on the whole trace set. The following three metrics are to evaluate the quality of classification results for each application (Cloud Computing web service in this report) class.

(ii) Precision: the ratio of True Positives over the sum of True Positives and False Positives or the percentage of flows that are properly attributed to a given application. True Positives is the number of correctly classified flows, False Positives is the number of flows falsely ascribed to a given application, and False Negatives is the number of flows from a given application that are falsely labeled as another application.

(iii) Recall: the ratio of True Positives over the sum of True Positives and False Negatives or the percentage of flows in an application class that are correctly identified.

(iv) F-measure: as a widely-used metric in information retrieval and classification, it considers both precision and recall in a single metric by taking their harmonic mean (2 × precision × recall) / (precision + recall). We use this metric to measure the per-application classification performance of machine learning algorithms.

(v) Classification speed: the number of classification decisions performed per second.

In this report, we use all above metrics except for the classification speed metric.

Chapter 3: Test Environment Setup

3.1 Hardware and Network Setup

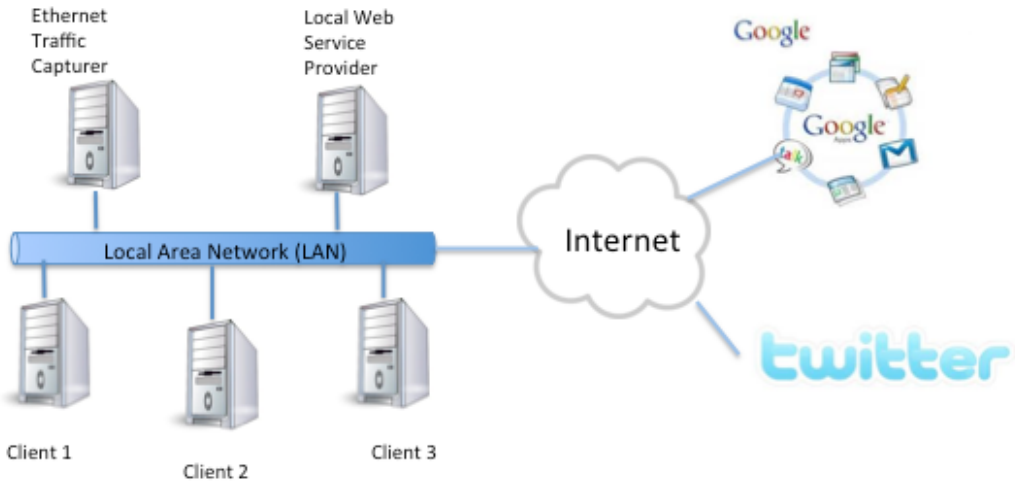

Figure 4: Test Environment Setup

The main constraints in the experiments are that (i) there is no existing traffic monitoring point where I can monitor a real Internet traffic with dominant cloud computing service presence; (ii) Even there is such a traffic monitoring point in the real Internet, it will be impossible to know the ground truth – which Cloud Computing Service a group of traffic flows belong to – due to end-to-end traffic encryption. Thus, I will collect all the test data through a simulated environment, where it only generates Cloud Computing services and provides visibility to the source of service invocation. The test environment includes the following features:

• One Ethernet traffic capturer that can monitor and capture all Ethernet traffic in/out the servers belong to the same Local Area Network (LAN).

• Three local hosts (denoted as client 1, client 2 and client 3 in above figure) that serve as Cloud Computing service clients.

• The Local Area Network (LAN) provides access to Internet. All local Cloud Computing clients have access to Cloud Computing services exposed by Google and Twitter.

• A local Cloud Computing Service simulator provides special types of service that use fire-and-forget message exchange pattern and subscribe-and-publish message exchange pattern. The main reason of having a local service provider instead of real Internet services is because the subscribe-and-publish message exchange services require each Cloud Computing client to have static Internet IP address, which is not available in my test environment.

3.2 Cloud Computing Services

Below table describes all the Cloud Computing services tested in the experiments.

Provider	Service Name	Description
Twitter	Twitter OAuth	Twitter application authentication.
	Twitter Timeline	Timelines are collections of Tweets, ordered with the most recent first.
	Twitter Help	These methods assist you in working & debugging with the Twitter API.
	Twitter Tweets	Tweets are the atomic building blocks of Twitter, 140-character status updates with additional associated metadata.
	Twitter Followers	Users follow their interests on Twitter through both one-way and mutual following relationships.

Table 1: Cloud Computing Services Tested

Provider	Service Name	Description
Google	Google OAuth	Google application authentication.
	Google Books	Search the complete index of Google Books and integrate with its social features.
	Google Document	Enable user to view and update your list of Google Documents.
	Google Maps	Use URL requests to access geocoding, directions and etc.
	Google URL Shortener	Create, inspect, and manage goo.gl short URLs from your desktop, mobile, or web application.
Local Simulator	Local Bank	MEP: Request-response. Random response length: 0~150 bytes.
	Local House	MEP: Fire-and-forget. Random message length: 0~350 bytes
	Local Movie	MEP: Subscribe-and-publish.
	Local Music	MEP: Request-response. Random response length: 0~350 bytes.
	Local Photo	MEP: Fire-and-forget.
	Local Stock	MEP: Subscribe-and-publish. Random message length: 0~350 bytes
	Local Toy	MEP: Request-response. Random response length: 0~1350 bytes.
	Local Travel	MEP: Request-response. Random response length: 0~1850 bytes.
	Local Video	MEP: Fire-and-forget.
	Local Weather	MEP: Subscribe-and-publish.

Table 1 continued: Cloud Computing Services Tested

3.3 Ground Truth

In order to train the classification algorithm, we need to know the ground truth – to which Cloud Computing service a given TCP packet belongs. Encrypted Cloud Computing traffic does not allow any visibility from traffic capturing node and the only visibility is from the client side or server side. In those experiments, all the service clients generate service logs and each log record contains start time, stop time, client IP address and destination. Then, the service log can be correlated to the data captured by traffic monitoring node based on timestamp and client IP address.

3.4 Captured Data Processing and Analysis

All the data captured are stored into a database and then preprocessed using SQL scripts. KNIME tool is used to analyze all preprocessed data. KNIME provides an implementation of C4.5 decision tree. Most of the techniques used in the decision tree implementation can be found in *C4.5 Programs for machine learning*, by J.R. Quinlan [15] and in *SPRINT: A Scalable Parallel Classifier for Data Mining*, by J. Shafer, R. Agrawal, M. Mehta [16].

All data collected from client 1 and client 2 are fed into C4.5 classification algorithm as training data and client 3 data are used to test the model output by the classification algorithm.

Chapter 4: Mining Result Evaluation

4.1 Mining Methods and Overall Accuracy

Below table shows all the mining methods used to analyze the data collected in the experiments and the related overall accuracy.

Method Number	Input Parameters	Incorrectly Classified	Correctly Classified	Overall Accuracy
Group 1	Individual TCP Packet Level Analysis			
1.1	Source IP, destination IP, protocol, length, source port, destination port	11266	0	0%
Group 2	Host Behavior - TCP Flow Analysis			
2.1	Source IP, destination IP, protocol, source port, destination port	9003	2263	20%
2.2	Source IP, destination IP, protocol, source port, destination port, length of Packet 1~10	197	154	44%
2.3	Source IP, destination IP, protocol, source port, destination port, length of Packet 1~10, direction of Packet 1~10	197	154	58%
2.4	Source IP, destination IP, protocol, source port, destination port, length of Packet 1~22	146	205	58%
2.5	Source IP, destination IP, protocol, source port, destination port, length of Application Data Packet 1~5	93	258	74%
2.6	Destination IP, destination port, length of Application Data Packet 1~5	70	281	80%

Table 2: Traffic Mining Methods and Overall Accuracy

Method Number	Input Parameters	Incorrectly Classified	Correctly Classified	Overall Accuracy
Group 3	Host Behavior - Connection Pattern Analysis			
3.1	Source IP, destination IP, protocol, source port, destination port, length of Application Data Packet1~5, related TCP session number	71	280	80%
3.2	Destination IP, destination port, length of Application Data Packet 1~5, related TCP session number	53	298	85%
Group 4	Message Exchange Pattern Analysis			
4.1	Destination IP, destination port, length of Application Data Packet 1~5, related TCP session number, MEP type	38	313	89%

Table 2 continued: Traffic Mining Methods and Overall Accuracy

The overall accuracy varies from 0% to 89% depending on the mining methods used. The results are analyzed in more detail below.

4.2 Individual TCP Packet Level Analysis

Group 1 method is based on analyzing individual TCP packet directly and attempts to find the correlation between TCP packet attributes (including source IP, destination IP, protocol, length, source port and destination port) and the Cloud Computing service class. Although, this method has the least overhead from real time data processing perspective, this method does not work at all against the test data captured in this experiment. We're not surprised to see that based on the discussion we had in the heuristics part (chapter 2.1.1.1). This method would yield a better output if each Cloud Computing service were associated with a different IP address, i.e., a service provider only provides one single Cloud Computing service.

4.3 Host Behavior - TCP Flow Analysis

Group 2 methods firstly aggregate individual TCP packets that share the same 5-tuple TCP connection attributes (Source IP, destination IP, protocol, source port and destination port) into a TCP session and attempt to look for the correlation between TCP session attributes and the Cloud Computing service class. A TCP session has many attributes including Source IP, destination IP, protocol, source port, destination port, direction of each packet, length of each packet and etc. Method 2.1 only uses basic 5-tuple TCP connection attributes and it yields a 20% overall accuracy. This method sets the baseline for this group. Then, we included the length of first 10 packets in method 2.2 and the overall increases to 44%. Method 2.3 shows direction of packet also is helpful. Method 2.4 uses more packet numbers and it helps too. Method 2.5 and 2.6 focus on achieving the same performance with minimum input parameters and we found that processing TLS Application data related packet statistics only is most efficient. And the source IP address and source port can also be skipped.

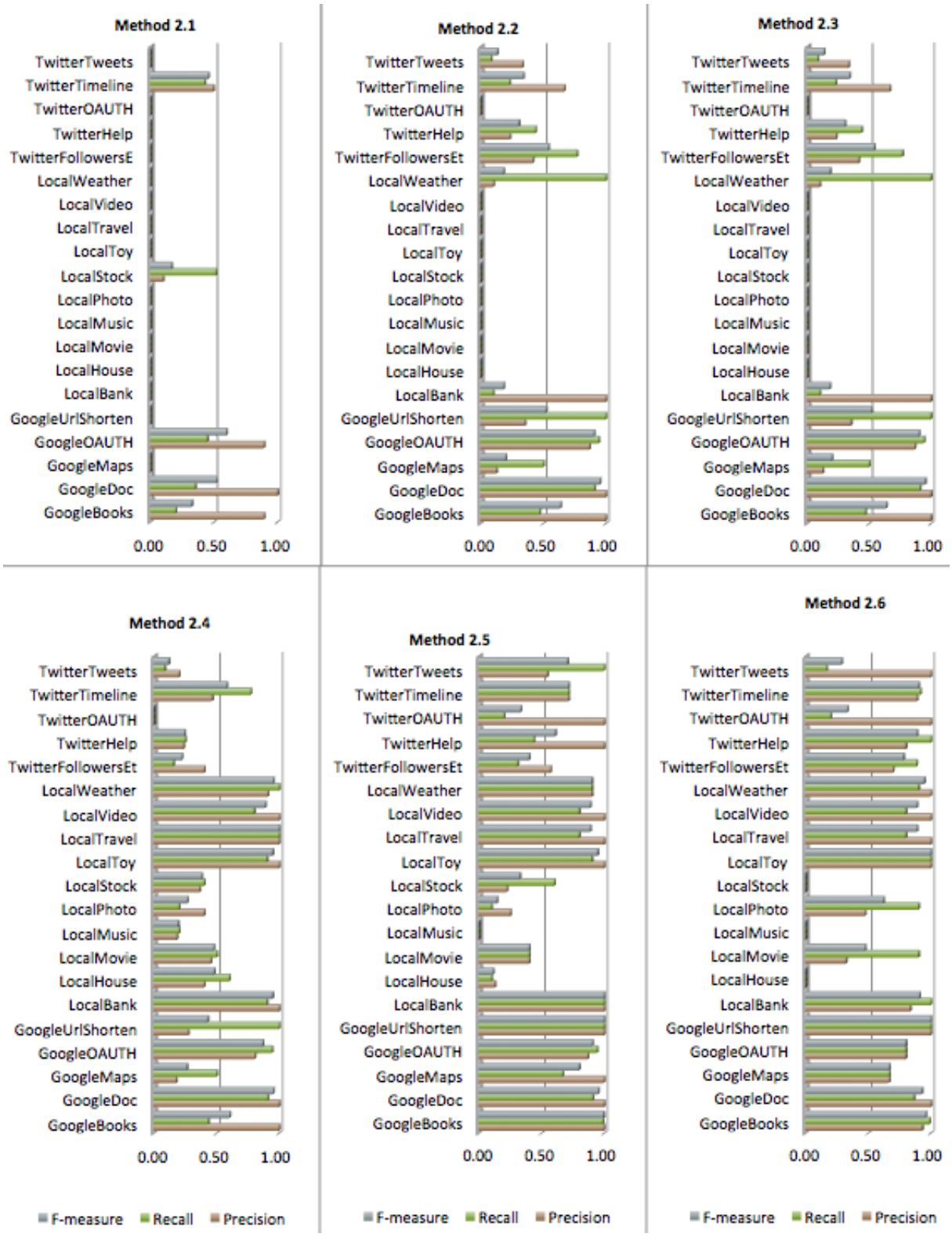

Figure 5: Mining Result Comparison - Group 2

Figure 5 shows detailed comparison among all the methods in Group 2 based on individual class level metrics: precision, recall and F-measure. Although method 2.6 yields best overall accuracy, it performs worse than method 2.4 for the simulated local services where the length of response data is purposely randomized. It suggests that processing more number of packets is helpful to handle services with random length message body.

4.4 Host Behavior - Connection Pattern Analysis

Group 3 methods further employ the connection patterns between clients and servers. Method 3.2 having better performance than method 3.1 shows that source IP address and port would obfuscate the classification result. Group 3 methods show increased performance over Group 2 methods by considering connection patterns.

4.5 Message Exchange Pattern Analysis

Group 4 methods use Cloud Computing specific feature, Message Exchange Pattern, in the classification. It has the highest overall accuracy among all the methods used in this testing.

Figure 6 shows that Method 4.1 has a significant classification performance increase than Method 3.2 for the simulated local web services where different Message Exchange Patterns (MEP) are applicable. Method 3.2 has very low performance in classifying several simulated services, including LocalStock service, LocalMusic service and LocalHouse service, because the length of related messages are similar to each other. Method 4.1 improves the situation by using MEP related information.

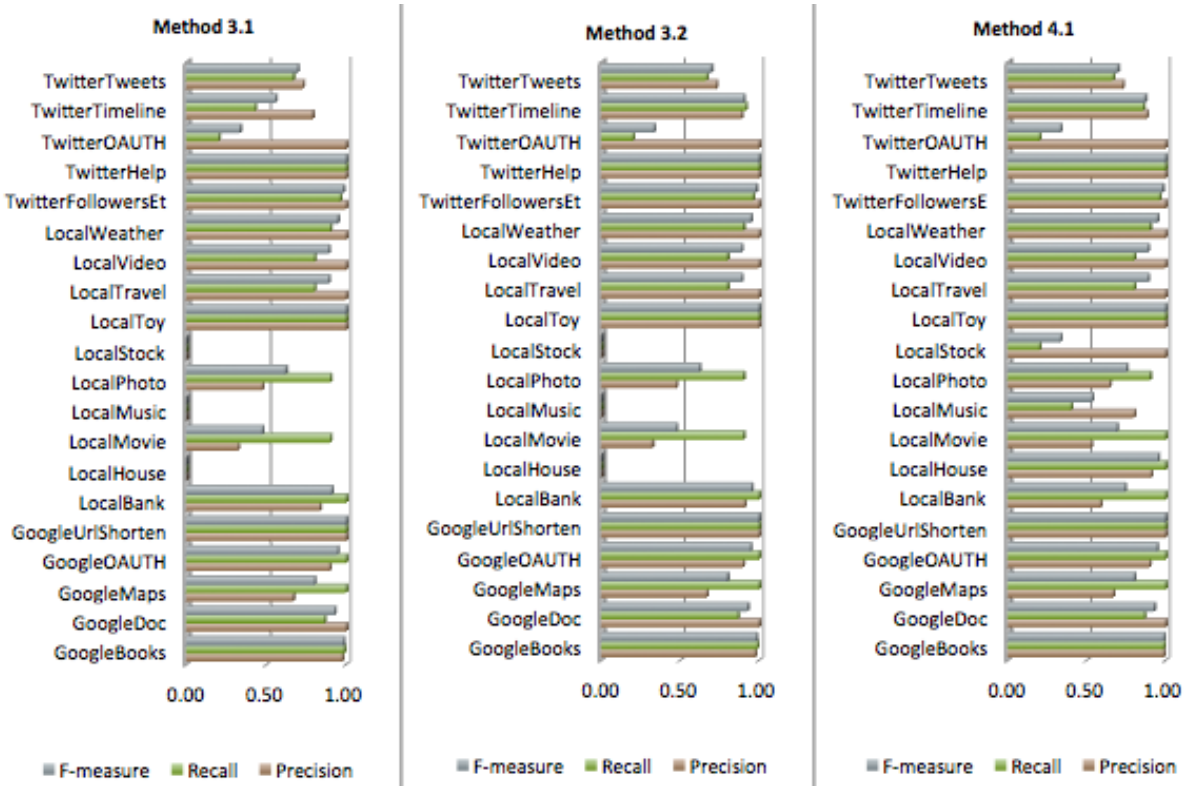

Figure 6: Mining Result Comparison – Group 3 & 4

Chapter 5: Conclusion

5.1 Summary

By combining TCP session level attributes, client and host connection patterns and Cloud Computing service Message Exchange Patterns (MEP), the method identified in this report yields 89% overall accuracy for classifying encrypted Cloud Computing traffic. This level of accuracy can be used in some practical business scenarios, but not all.

Analyzing individual TCP packet level attribute is seen as dysfunctional when a service provider provides more than one services. Individual packets must be aggregated into TCP sessions based on 5-tuple TCP connection attributes (Source IP, destination IP, protocol, source port and destination port) before applying the mining algorithm. Separating the statistics of TLS application related packets from normal TLS control (e.g. TLS handshake) related packets helps to achieve higher performance while dealing with less computing overhead. Client-server connection pattern contributes to the classification performance increase. And Cloud Computing service specific characteristics, Message Exchange Patterns (MEP) in this report, definitely helps improving the overall accuracy assuming the real world Cloud computing has a mixed used of all message exchange patterns.

5.2 Future Work

The work presented in this report is still far from the needs of a practical application. The accuracy needs to be further improved and also it needs to be studied from a real-time processing effectiveness perspective. Real Internet network load is a

mixture of traditional non-cloud-computing traffic and Cloud Computing service traffic. This method needs to be improved to adapt to such a heterogeneous environment.

Some thoughts about further improving the accuracy: A Cloud Computing service does not act alone. It enhances its own functionalities by collaborating with other services in the Internet. For example, a weather services may invoke a location service before sends out the weather content. A Cloud Computing service with complex logic may have more complex interactions with other services. This Service Mash-up Pattern (SMP) could be unique to a Cloud Computing service, thus can be used as a key input to the classification algorithm.

References

[1] Peter Mell and Tim Grance, 2009, The NIST Definition of Cloud Computing.

[2] Web Services Architecture, W3C Working Group Note 11, February 2004.

[3] Roy Thomas Fielding, 2000, Architectural Styles and the Design of Network-based Software Architectures

[4] Leonard Richardson & Sam Ruby, 2011, RESTful Web Services, ISBN: 9780596-52926-0

[5] Security Architecture of OSI, ITU-T Recommendation X.800, 1991

[6] T. Dierks and C. Allen, 1999, The TLS Protocol, Version 1.0, Network Working Group, IETF RFC 2246

[7] E. Rescorla, 2000, HTTP Over TLS, Network Working Group , IETF RFC 2818

[8] Antonio Martin, Carlos Leon, Felix Biscarri, 2010, Intelligent Integrated Management for Telecommunication Networks, International Journal of Advancements in Computing Technology, vol.2, no. 2, pp. 158-171.

[9] Thuy T.T. Nguyen, Grenville Armitage, A Survey of Techniques for Internet Traffic Classification using Machine Learning, Accepted 16 Nov 2007 for 4th edition 2008 of IEEE Communications Surveys and Tutorials

[10] Yeon-sup Lim, Hyun-chul Kim, Jiwoong Jeong, Chong-kwon Kim, Ted "Taekyoung" Kwon, Yanghee Choi, 2008, Internet Traffic Classification Demystified: On the Sources of the Discriminative Power

[11] Hengjie GU, Shunyi ZHANG, Xiaozhen XUE, 2011, Encrypted Internet Traffic Classification Method based on Host Behavior,

[12] S. Deering, R. Hinden, 1998, Internet Protocol, Version 6 (IPv6) specification, IETF RFC 2460

[13] S. Kent, K. Seo, 2005, Security Architecture for the Internet Protocol, IETF RFC 4301

[14] Thomas Erl, 2006, Service-Oriented Architecture Concepts, Technology, and Design, ISBN 0-13-18585809, Fifth Printing

[15] J.R. Quinlan, 1993, C4.5 Programs for machine learning

[16] J. Shafer, R. Agrawal, M. Mehta, Sep 1996, SPRINT: A Scalable Parallel Classifier for Data Mining, Proc. of the 22th Int'l Conference on Very Large Databases, Mumbai (Bombay), India.

[17] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, 2006, "Traffic classification on the fly," ACM Special Interest Group on Data Communication (SIGCOMM) Computer Communication Review, vol. 36, no. 2.

[18] T. Nguyen and G. Armitage, November 2006, "Training on multiple sub-flows to optimize the use of Machine Learning classifiers in real-world IP networks," in Proc. IEEE 31st Conference on Local Computer Networks, Tampa, Florida, USA.