Use Case Analysis and Potential Bandwidth Optimization Methods for Encrypted Traffic

Jianjie You, Hanyu Wei, Huaru Yang
Huawei

Abstract:
Increasing encryption on the internet poses traffic management challenges to network operators. In this paper we analyze the impacted services and categorize them into three types, i.e. low/middle/high-level dependence services. We propose two potential bandwidth optimization methods for low/middle-level dependence services. The objective of this paper is to point out the impacts of encrypted traffic and the challenges for bandwidth optimization methods.

1. Introduction

Encryption of internet traffic is to prevent pervasive monitoring and protect customer privacy. Historically, Secure Sockets Layer (SSL) / Transport Layer Security (TLS) were earlier used in financial services to encrypt a subset of Internet traffic, especially financial transactions. However, the shift away from unencrypted traffic towards encrypted traffic is accelerating in recent years [1] due to concerns about privacy. Google offered end-to-end encryption for Gmail since 2010, and switched all searches over to HTTPS in 2013. YouTube traffic is carried via HTTPS (or QUIC) since 2014. Also, the Snowden revelations seem to cause an upward surge in encrypted traffic [2]. A large number of operators began requiring encryption for all XMPP traffic in May 2014 [3].

However, the prevalence of encryption impacts current network services, such as policy control, load balancing, etc. The network services may be less efficient or totally unavailable in the case of fully encrypted traffic. Through our analysis of impacted services in the case of encrypted traffic, we find that the impacted services can be categorized into three types based on the level of dependence to content visibility:

A: Low-level dependence
A service that is low-level dependent on the content visibility means the service can be effective providing with flow type (e.g. stream ID) rather than parsing the content itself. The typical service of low-level dependence is load balancing, which will be discussed in section 2.1.

B: Middle-level dependence
A service that is middle-level dependent on the content visibility means the service can be effective providing with access metadata (e.g. domain name, URI) besides flow type rather than parsing the content itself entirely. Through the metadata different access features can be distinguished, thus appropriate actions could be enforced based on these features. For example, illegal websites can be filtered. The typical service of middle-level dependence is parental controls, which will be discussed in section 2.2.

C: High-level dependence
A service that is high-level dependent on the content visibility means the service can be effective requiring analysis of content itself, even interaction procedure. The typical service of high-level

dependence is video caching, which usually requires user access behavior and detailed video content (e.g. encoding format). In the case of encrypted traffic, this kind of service will not be available.

2. Typical Use Case Analysis

2.1 Low-level Dependence Service

Low-level dependence service doesn't require clear text access to the application layers of interest, but rather limited information. Equal-cost multi-path (ECMP) is a typical example. ECMP is a routing technique for routing packets along multiple paths of equal cost. In practice, many implementations use the 5-tuple {source IP address, destination IP address, protocol number, source port number, and destination port number} as input keys to the hash function, to maximize the probability of evenly sharing traffic over the equal cost paths. However, including transport-layer information as input keys to a hash may be a problem for encrypted traffic. In the case of encrypted HTTP/2, HTTP/2 is designed to have fewer, longer-lived connections, due to its multiplexing. This provides a negative effect on load balance. For the unencrypted traffic in HTTP/2, Stream Identifier could be used as an additional input key besides 5-tuple; thus, the performance would return to the original one based on 5-tuple before multiplexing. However, when the traffic is encrypted, little flow information is visible to the network. In the case of encrypted HTTP/2, still hashing using 5-tuple would lead to low load balancing performance, which may aggravate the link congestion.

We compare the performance of load balance between HTTP/2 multiplexing used before and after. Assigning m streams uniformly at random to L channels, the number of streams assigned to a given channel has a Binomial distribution with parameters:
n = m - number of streams; and
p, where 1/p = L is the number of channels
The standard deviation rate [4] is

$$\partial = \frac{\sqrt{\mathrm{np}(1-\mathrm{p})}}{\mathrm{n} \times \mathrm{p}} \times 100\%$$

For example, before multiplexing, assume p=10% (i.e. 10 channels), n=10000; then ∂=3%. After multiplexing, assume p=100% (i.e. 1 channel), n=1000; then ∂=9%. As we can see, the deviation rate is increasing with p increasing and n decreasing. Usually, operators requires standard deviation rate less than 5%.

2.2 Middle-level Dependence Service

Middle-level dependence service requires part of content or features in order to make it effective. This kind of services includes parental controls, traffic filtering, etc. We take parental controls as a typical example discussed below.

Parental controls are a common service provided by operators. This service helps parents manage how children's access to the Internet. One popular type of parental controls is content

filtering that limits access to internet content, such as blocking dangerous sites.

Service provider usually needs keywords, such as domain name, objects' URI, destination address, and special keywords in the content, to predefine access control rules. Once the traffic matching the rules, the corresponding action will be executed.

Parental controls can be deployed using local filtering software or external gateway. The filtering software is installed locally within the computer for blocking websites; and the external gateway does traffic monitoring according to access control rules in the network. However the filtering software can be easily bypassed by booting up the computer or even by using new web browser software. By contrast, parental controls using external gateway is more robust. However, the use of session encryption to access application-layer semantics will limit the ability to use session data to ensure access control. Session encryption at the application level, prevents access to URL, keywords etc. Thus, the access control rules based on the information above are not effective any more. For example, if an endpoint accesses Google through a proxy and uses encryption, then all the data accessed through the proxy is not visible except to the proxy. So in order to maintain parental controls, the feature information must be provided.

2.3 High-level Dependence Service

High-level dependence service (e.g. Deep Packet Inspection, DPI) requires analysis of content itself and user transactions information in order to make it effective. Encryption makes this kind of services unavailable any more.

3. Potential Bandwidth Optimization Methods

3.1 Legacy Protocol Extension

Regarding to the low-level dependence services, existing protocols could be extended in order to carry flow type, for example, extending TLS header (Figure 1):

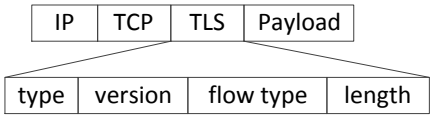

Figure 1: TLS header extension

Though TLS SNI (Server Name Indication) may be used to carry necessary information for low-level dependence services, the per-domain nature of SNI [5] may not reveal the specific service or media type being accessed. So extending a new field that indicates flow type into the TLS header seems a potential method in the case of encrypted traffic.

3.2 New Substrate Protocol

New substrate protocols (Figure 2) over existing transport layers, such as UDP, TCP, are considered to carry flow information in order to make middle-level dependence service effective.

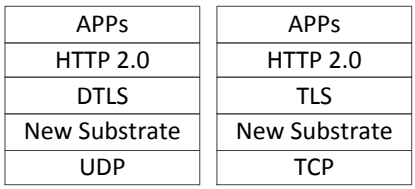

Figure 2: New Substrates

Developing UDP-based substrate protocols to enable transport evolution is a hot topic in IETF recently. The QUIC protocol from Google falls into this space; however, QUIC is not aiming to solve the encryption issues. One major issue with UDP-based substrate is middleboxes may block UDP or limit rate. SPUD-like UDP-based substrate could be a potential method to allow traffic management while using transport protocols. How middleboxes trust the information exposed by the endpoints should be considered.

However today's Internet is full of middleboxes that may interfere with the information sent in IP packets and TCP segments. "Is it still possible to extend TCP?" [6] shows the limitation imposed on TCP extensions by middleboxes behaviors, such as TCP options removed or updated, the source and destination port numbers translated by NATs. Though we can still extend TCP to support middle-level dependence services, extensions are very constrained as it needs to take into account middleboxes behaviors.

4. Conclusion

Under the traffic encryption, the operator services are concluded into three categories: low/middle/high-level dependence services. Regarding to the low/middle-level dependence services, the challenges for potential traffic management methods for encrypted traffic are analyzed. Furthermore, possible IETF standardization work (i.e. legacy protocol extensions and new substrates) is explored in order to solve the conflict between user privacy and traffic management.

5. Reference

[1]. I-D.mm-wg-effect-encrypt, Effect of Ubiquitous Encryption, March 7, 2015
[2]. RFC7258, Pervasive Monitoring Is an Attack, May 2014
[3]. "XMPP switches on mandatory encryption" (http://lwn.net/Articles/599647/)
[4]. Eli Upfal, Description and Analysis of the Current Load Balancing Procedure, August 28, 2013.
[5]. I-D.smith-encrypted-traffic-management, Network management of encrypted traffic, June 10, 2015.
[6]. Michio Honda, Yoshifumi Nishida, Costin Raiciu, Adam Greenhalgh, Mark Handley, Hideyuki Tokuda, Is it Still Possible to Extend TCP, IMC'11, November 2–4, 2011, Berlin, Germany.