CCIT REPORT #770 July 2010

PACK: Speculative TCP Traffic Redundancy Elimination

Eyal Zohar
Electrical Engineering Department
Technion
eyalzo@tx.technion.ac.il

Osnat (Ossi) Mokryn
School of Computer Science
Tel Aviv College
ossi@mta.ac.il

Israel Cidon
Electrical Engineering Department
Technion
cidon@ee.technion.ac.il

***Abstract*—Eliminating redundant network traffic became an important task with the volume increase of large files and rich media content. Consequently, many commercial traffic redundancy elimination (TRE) middleboxes, were placed at WAN and Internet access points.**

However, recent studies have shown that the majority of traffic redundancy results from end-to-end exchanges. Moreover, the penetration of laptops and smart phones has detached clients from specific access middleboxes. Consequently, there is a rising need for a universal, software based, end-to-end transport level TRE.

As many central services such as email and streaming video may use these new capabilities, it is important to minimize the overhead and latency increase associated with the TRE operations.

We present a novel low latency, low overhead, universal TRE mechanism, termed PACK. PACK is designed as a TCP extension and supports all applications built over TCP.

The main idea for reducing the latency and server overheads is PACK's receiver based speculative operation replacing the common sender based approach. The receiver sends data predictions to the sender which in turn moves to computational expensive actions only when these predictions are correct.

Consequently, PACK is best suited for highly loaded servers. Other benefits of PACK are its low latency and buffering requirements.

I. Introduction

TCP/IP traffic at both intranets and the Internet exhibits a significant amount of redundancy and replication. Traffic redundancy results from common users activities such as repeatedly accessing and modifying the same information (documents, data, web and video), and from communicating and sharing of information among multiple users.

Traffic redundancy at enterprise offices was found to vary between 20% to 60% [1][2][3], exacerbating the need for traffic redundancy elimination (TRE), as a mean to reduce network loads and costs and to speed up communication and applications.

A major progress in handling redundant data, followed several seminal papers that set the foundation of modern TRE [4], [1], [5]. In such techniques, the sender and the receiver compare signatures of data chunks, parsed according to the data content, prior to their transmission, possibly eliminating their transmission.

In [5] files are divided into chunks using the method described in [4], by computing a 48-byte Rabin fingerprint [6] for each byte and defining as the beginning of the chunk the byte for which the 12 least significant bits take a predefined value. A SHA-1 signature is calculated for each chunk. These TRE algorithms require that both sides perform file indexing and signature calculations.

Subsequently, TRE techniques have been explored at both the industry and the academic community to eliminate the redundant content and to significantly improve the network speed and efficiency. Most commercial TRE solutions involve the deployment of two or more middleboxes at both the intranet entry points of data centers and branch offices, eliminating repetitive traffic due to sharing, and preventing the exchange of redundant data between clients and data centers. Examples of commercial TRE solutions vendors are Cisco [7], Riverbed [8], Quantum [9], Juniper [10], Bluecoat [11], Expand Networks [12] and F5 [13].

A recent work [2] reports that in an enterprise environment 75%-90% of the redundancy captured by the shared middleboxes stemmed from an end-to-end redundancy (i.e. data redundancy in the traffic exchanged by a single server and a single client). Consequently, it is claimed that an end-to-end (software based) TRE solution can achieve most of the bandwidth savings of the TRE middleboxes, and offer higher savings for small to medium organizations, home offices and end users. The main advantages of an end-to-end solution are the reduction of cost, space and maintenance due to the elimination of the middlebox hardware. It also enables the freedom to move from incompatible proprietary vendor solutions to a standard protocol stack, and to cope with end-to-end encryption methods (e.g., IPsec). Finally, the vast use of mobile Wi-Fi laptops and smart phones eliminate the association between clients and middleboxes. Therefore, it becomes evident that an application transparent, end-to-end, standard TRE can benefit the emerging data intensive services and applications.

We believe that an end to end, software based TRE should meet the following desirable properties:

1) Standard: The TRE standard needs to work across all server and client platforms and operating systems. This will enable servers to reduce redundant traffic, regardless of the client nature and location.
2) Application independent: The TRE should support most applications that transmit redundant information. Similar data may be observed across different applications, e.g., mail attachments may repeat data transmitted by the file system, FTP or web browsing. This calls for the