ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization *Vol.3, Special Issue 4, April 2015*

National Conference On Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -15)

Organized by

Dept. of ECE, Annamacharya Institute Of Technology & Sciences, Rajampet, Andhra Pradesh-516126, India held on 28th February 2015

Functional Implementation of PACK in Cloud Based Environment

B.Mounika

Student, Dept. of CSE, New Boyanapalli, Rajampet, Kadapa(dist), Andhra Pradesh, India

ABSTRACT: Generally speaking PACK is based on a novel TRE technique, which allows the client to use newly received chunks such as pay-as-you-go service model, known also as usage-based pricing. Prediction-Based Cloud Bandwidth and Cost Reduction System. How to reduce Smartphone traffic volume by 0%?.This aspect can be made with this cloud based environments, traffic redundancy elimination is an very important aspect of pack. It shows a fully functional PACK implementation, transparent to all TCP-based applications and network devices. Finally, PACK benefits for cloud users, using traffic traces from various sources.

KEYWORDS: Caching, cloud computing, network optimization, traffic redundancy elimination.

I. INTRODUCTION

CLOUD computing offers its customers an economical and convenient *pay-as-you-go* service model, known also as *usage-based pricing*. Cloud customers pay only for the actual use of computing resources, storage, and band-width, according to their changing needs, utilizing the cloud's scalable and elastic computational capabilities. In particular, data transfer costs (i.e., bandwidth) is an important issue when trying to minimize costs. Consequently, cloud customers, applying a judicious use of the cloud's resources, are motivated to use various traffic reduction techniques, in particular traffic Redundancy elimination (TRE), for reducing bandwidth costs.

Traffic redundancy systems from common end-users' activities, such as repeatedly accessing, downloading, uploading (i.e., backup), distributing, and modifying the same or similar information items (documents, data, Web, and video). TRE is used to eliminate the transmission of redundant content and, there-fore, to significantly reduce the network cost. In most common TRE solutions, both the sender and the receiver examine and compare signatures of data chunks, parsed according to the data content, prior to their transmission. When redundant chunks are detected, the sender replaces the transmission of each redundant chunk with its strong signature. Commercial TRE solutions are popular at enterprise networks, and involve the deployment of two or more proprietary- protocol, state synchronized middle-boxes at both the intranet entry points of data centers and branch offices, eliminating repetitive traffic between them.

While proprietary middle-boxes are popular point solutions within enterprises, they are not as attractive in a cloud environment. Cloud providers cannot benefit from a technology whose goal is to reduce customer bandwidth bills, and thus are not likely to invest in one. The rise of "on -demand" work spaces, meeting rooms, and work from home solutions detaches the workers from their offices. In such a dynamic work environment, fixed point solutions that require a client-side and a server -side middle-box pair become ineffective.

Clearly, a TRE solution that puts most of its computational effort on the cloud side may turn to be less cost-effective than the one that leverages the combined client-side capabilities. Given an end-to-end solution, we have found through our experiments that sender-based end-to-end TRE solutions add a considerable load to the servers, which may eradicate the cloud cost saving addressed by the TRE in the first place. Our experiments further show that current end-to-end solutions also suffer from the requirement to maintain end-to-end synchronization that may result in degraded TRE efficiency.

It presents a novel receiver-based end-to-end TRE solution that relies on the power of predictions to eliminate redundant traffic between the cloud and its end-users. In this solution, each receiver observes the incoming stream and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization *Vol.3, Special Issue 4, April 2015*

National Conference On Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -15)

Organized by

Dept. of ECE, Annamacharya Institute Of Technology & Sciences, Rajampet, Andhra Pradesh-516126, India held on 28th February 2015

tries to match its chunks with a previously received chunk chain or a chunk chain of a local file. Using the long term chunks' meta-data information kept locally, the receiver sends to the server predictions that include chunks' signatures and easy-to-verify hints of the sender's future data. The sender first examines the hint and performs the TRE operation only on a hint- match. The purpose of this procedure is to avoid the expensive TRE computation at the sender side in the absence of traffic redundancy. When redundancy is detected, the sender then sends to the receiver only the ACKs to the predictions, instead of sending the data.

Offloading the computational effort from the cloud to a large group of clients forms a load distribution action, as each client processes only its TRE part. The receiver-based TRE solution addresses mobility problems common to quasi-mobile desktop/ laptops computational environments. One of them is cloud elasticity due to which the servers are dynamically relocated around the federated cloud, thus causing clients to interact with multiple changing servers. Another property is IP dynamics, which compel roaming users to frequently change IP addresses. In addition to the receiver-based operation, we also suggest a hybrid approach, which allows a battery-powered mobile device to shift the TRE computation overhead back to the cloud by triggering a sender-based end-to-end TRE.

II. RELATED WORK

The model can be implemented in client-server aspect. Several TRE techniques have been explored in recent years. A protocol independent TRE was proposed. The paper describes a packet-level TRE, utilizing the algorithms presented. Assume that the routers are equipped with data caches, and that they search those routes that make a better use of the cached data.

EndRE is a sender-based end-to-end TRE for enterprise networks. It uses a new chunking scheme that is faster than the commonly used Rabin fingerprint, but is restricted to chunks as small as 32–64 B. Unlike PACK, EndRE requires the server to maintain a fully and reliably synchronized cache for each client. To adhere with the server's memory requirements, these caches are kept small (around 10 MB per client), making the system inadequate for medium-to-large content or long-term redundancy. EndRE is server-specific, hence not suitable for a CDN or cloud environment.

A sender-based end-to-end TRE for enterprise networks. It uses a new chunking scheme that is faster than the commonly used Rabin fingerprint, but is restricted to chunks as small as 32–64 B. Unlike PACK, EndRE requires the server to maintain a fully and reliably synchronized cache for each client. To adhere with the server's memory requirements, these caches are kept small (around 10 MB per client), making the system in-adequate for medium-to-large content or long-term redundancy. EndRE is server-specific, hence not suitable for a CDN or cloud environment.

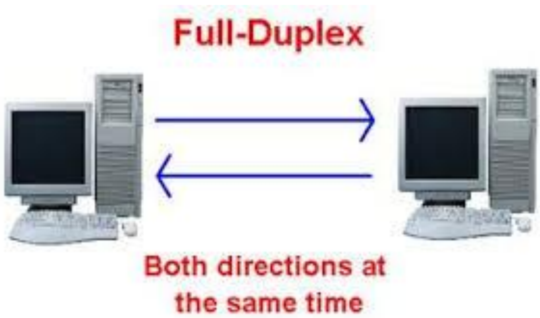

Fig .1. From stream to chain

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.3, Special Issue 4, April 2015

National Conference On Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -15)

Organized by

Dept. of ECE, Annamacharya Institute Of Technology & Sciences, Rajampet, Andhra Pradesh-516126, India held on 28th February 2015

III. PROPOSED ALGORITHM

a. Design for PACK Considerations:

- The stream of data received at the PACK receiver is parsed to a sequence of variable-size, content-based signed chunks similar to.
- The chunks are then compared to the receiver local storage, termed *chunk store*.
- If a matching chunk is found in the local chunk store, the receiver retrieves the sequence of subsequent chunks, referred to as a *chain*, by traversing the sequence of LRU chunk pointers that are included in the chunks' metadata.

b. *Description of the Proposed Algorithm:*

a. Receiver Chunk Store

PACK uses a new *chains* scheme, described in Fig. 1, in which chunks are linked to other chunks according to their last received order. The PACK receiver maintains a *chunk store*, which is a large size cache of chunks and their associated metadata. Chunk's metadata includes the chunk's signature and a (single) pointer to the successive chunk in the last received stream containing this chunk. Caching and indexing techniques are employed to efficiently maintain and retrieve the stored chunks, their signatures, and the chains formed by traversing the chunk pointers.

Unlike IP-level TRE solutions that are limited by the IP packet size, PACK operates on TCP streams and can therefore handle large chunks and entire chains. Although our design permits each PACK client to use any chunk size, we recommend an average chunk size of 8kB.

b. Receiver Algorithm

Upon the arrival of new data, the receiver computes the respective signature for each chunk and looks for a match in its local chunk store. If the chunk's signature is found, the receiver determines whether it is a part of a formerly received chain, using the chunks' metadata. If affirmative, the receiver sends a prediction to the sender for several next expected chain chunks. The prediction carries a starting point in the byte stream (i.e., offset) and the identity of several subsequent chunks (PRED command).

Proc. 1: Receiver Segment Processing

```
if segment carries payload data then
    calculate chunk
    if reached chunk boundary then
        activate predAttempt()
    end if
else if PRED-ACK segment then
    processPredAck()
    activate predAttempt()
end if
```

c. Sender Algorithm

When a sender receives a PRED message from the receiver, it tries to match the received predictions to its buffered (yet to be sent) data. For each prediction, the sender determines the corresponding TCP sequence range and verifies the hint. Upon a hint match, the sender calculates the more computationally intensive SHA- 1 signature for the predicted data range and compares the result to the signature received in the PRED message. Note that in case the hint does not match, a computationally expansive operation is saved. If the two SHA-1 signatures match, the sender can safely assume that

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization *Vol.3, Special Issue 4, April 2015*

National Conference On Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -15)

Organized by

Dept. of ECE, Annamacharya Institute Of Technology & Sciences, Rajampet, Andhra Pradesh-516126, India held on 28th February 2015

the receiver's prediction is correct. In this case, it replaces the corresponding outgoing buffered data with a PRED-ACK message.

d. Wire Protocol

In order to conform with existing firewalls and minimize overheads, we use the TCP Options field to First, both sides enable the PACK option during the initial TCP handshake by adding a *PACK permitted* flag (denoted by a bold line) to the TCP Options field. Then, the sender sends the (redundant) data in one or more TCP segments, and the receiver identifies that a currently received chunk is identical to a chunk in its chunk store. The receiver, in turn, triggers a TCP ACK message and includes the prediction in the packet's Options field. Last, the sender sends a confirmation message (PRED-ACK) replacing the actual data.

IV. PSEUDO CODE

Step 1: Generate all the possible payload data.
Step 2: Calculate the chunks for each node of each route using eq. (1).
Step 3: Check the below condition for each user sen the data is available to transmit the packet.

```
     if (payload< =chunk)
          generate PredAttempt message
     else
              generate Pred_ACK to user
              activate PredAttempt message
     endif
```

Step 4: now,sender and and receiver can sending data normally.
Step 5: End.

V. SIMULATION RESULTS

The simulation studies involve the deterministic *Traffic Traces:* obtained a 24-h recording of traffic at an ISP's 10-Gb/s PoP router, using a 2.4-GHz CPU recording machine with 2 TB storage (4 500 GB 7 200 RPM disks) and 1-Gb/s NIC,filtered YouTube traffic using deep packet inspection and mirrored traffic associated with YouTube servers IP addresses to our recording device.Measurements show that YouTube traffic accounts for 13% of the total daily Web traffic volume of this ISP. The recording of the full YouTube stream would require 3 times network and disk write speeds. Therefore,it can isolated 1/6 of the obtained YouTube traffic, grouped by the video identifier (keeping the redundancy level intact) using a programmed load balancer that examined the upstream HTTP requests and redirected downstream sessions according to the video identifier that was found in the YouTube's URLs, to a total of 1.55 TB. For accurate reading of the true redundancy,it can filtered out the client IP addresses that were used too intensively to represent a single user and were assumed to represent a NAT address.Note that YouTube's video content is not cacheable by standard Web proxies since its URL contains private single-use tokens changed with each HTTP request. Moreover, most Web browsers cannot cache and reuse partial movie downloads that occur when end-users skip within a movie or switch to another movie before the previous one ends.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization *Vol.3, Special Issue 4, April 2015*

National Conference On Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -15)

Organized by

Dept. of ECE, Annamacharya Institute Of Technology & Sciences, Rajampet, Andhra Pradesh-516126, India held on 28th February 2015

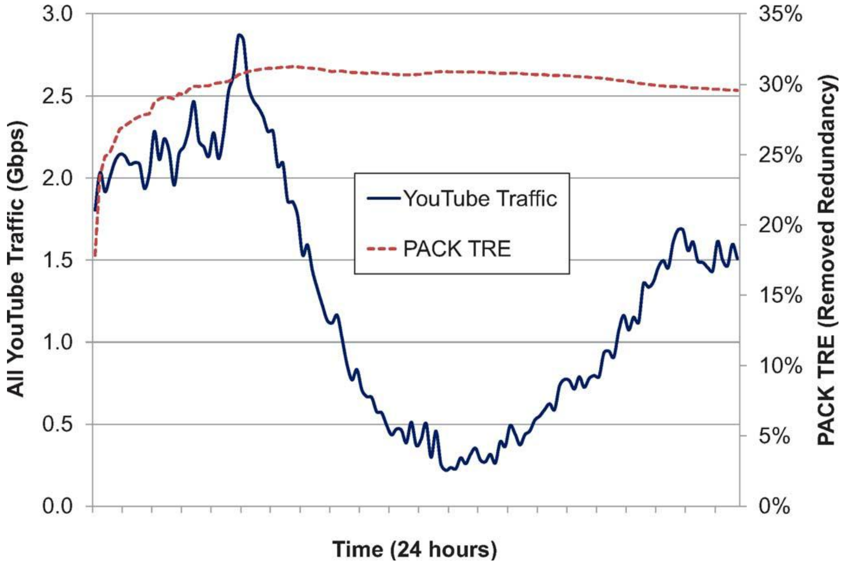

Fig. 2. ISP's YouTube traffic over 24 h, and PACK redundancy elimination ratio with this data.

VI. CONCLUSION AND FUTURE WORK

Conclusion for PACK, a receiver-based, cloud-friendly, end - to-end TRE that is based on novel speculative principles that reduce latency and cloud operational cost. PACK does not require the server to continuously maintain clients' status, thus enabling cloud elasticity and user mobility while preserving long-term redundancy. Moreover, PACK is capable of eliminating redundancy based on content arriving to the client from multiple servers without applying a three-way handshake.

REFERENCES

[1] N. T. Spring and D. Wetherall, "A protocol-independent technique for eliminating redundant network traffic," in *Proc. SIGCOMM*, 2000, vol. 30, pp. 87–95.
[2] A. Muthitacharoen, B. Chen, and D. Mazières, "A low-bandwidth net-work file system," in *Proc. SOSP*, 2001, pp. 174–187.
[3] E. Lev-Ran, I. Cidon, and I. Z. Ben-Shaul, "Method and apparatus for reducing network traffic over low bandwidth links," US Patent 7636767, Nov. 2009.
[4] S. Mccanne and M. Demmer, "Content-based segmentation scheme for data compression in storage and transmission including hierarchical segment representation," US Patent 6828925, Dec. 2004.
[5] R. Williams, "Method for partitioning a block of data into sub blocks and for storing and communicating such subblocks," US Patent 5990810, Nov. 1999.
[6] Juniper Networks, Sunnyvale, CA, USA, "Application acceleration," 1996 [Online]. Available: http://www.juniper.net/us/ en/products-services/application-acceleration/
[7] Blue Coat Systems, Sunnyvale, CA, USA, "MACH5," 1996 [Online]. Available: http://www.bluecoat.com/products/mach5
[8] R. Williams, "Method for partitioning a block of data into sub blocks and for storing and communicating such subblocks," US Patent 5990810, Nov. 1999.
[9] Juniper Networks, Sunnyvale, CA, USA, "Application acceleration," 1996 [Online]. Available: http://www.juniper.net/us/ en/products-services/application-acceleration/
[10] Blue Coat Systems, Sunnyvale, CA, USA, "MACH5," 1996 [Online]. Available: http://www.bluecoat.com/products/mach5

BIOGRAPHY

B.Mounika is a Post Graduate student in the CSE Department, College of Annamacharya institute Of Technology & Sciences,New Boyanapally, Rajampet. She received bachelor degree in 2012 from MKITW,New Boyanapalli, Rajampet, kadapa district,Andhra pradesh, India. Her research interests are Computer Networks (wireless Networks),Big Data.