RESEARCH ARTICLE OPEN ACCESS

An Implementation of Improving the Service Accessibility for Cloud Users by Using Traffic Redundancy Elimination (TRE) Mechanism with PACK Technique

Ms. R.Poorvadevi[1]. A.Lakshmidurga[2], Ms.D.SaiVijitha[3], Ms. J.Praseeda[4]
Assistant Professor[1] UG Scholar[2, 3 & 4]
Department of Computer Science and Engineering
SCSVMV University Kanchipuram- 631 561
TamilNadu - India

ABSTRACT

Nowadays, people are majorly doing their job and tasks in a quick response manner. Everything is manipulated and processed with the short time span. So, all the users are depending on the web to accomplish their functions. So, cloud is one which is offering enormous amount of services to its dependent users. However, we have all these services as a resource form; still CSP cannot provide the enough bandwidth capacity to its clients. This proposed model is, traffic redundancy elimination system will helps to analyze sufficient bandwidth capacity and focusing on cost-cutting systems in cloud environment. In this approach, three different features are used such as transparent billing system, service initiation and service termination (this alert is send as an information to the users in the form of messages and e-mail, once the service is activated and de-activated respectively).The PACK can be a proposed approach to avoid the traffic problem and also to find out the redundancy information's about the users and their service consumption and so on. Through this model (TRE) we can achieve the solution for traffic problem and reduce the service cost which is offered by cloud service provider.

Keywords:- PACK, chunks, TRE, bandwidth, middle box, cost reduction, CSP

I. INTRODUCTION

In the recent years cloud computing is achieving a high-demand as the number of cloud users are increasing day-by-day. The major success of cloud is due to its customer relationship as it gives the ability to use services on demand with a pay-as-you go pricing model for the various services being used, which has proved convenient in many respects. People started to prefer cloud computing due to its low cost and high flexibility. Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local Data storage and maintenance. Traffic redundancy and elimination is an effective approach for the reduction of cost in cloud. 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. Current end-to-end TRE solutions are sender-based. In the case where the cloud server is the sender, these solutions require that the server continuously maintain client's status. We now show here that cloud elasticity requires for a new TRE solution. First, the major actions that occur in server-side process and data migration environment are load balancing and power optimizations, in which TRE solutions that require full synchronization between the server and the client are hard to accomplish or may lose efficiency due to lost synchronization.

Second, the popularity of rich media that consume high bandwidth motivates content distribution network (CDN) solutions, in which the service point may change for the fixed and remote users in accordance to the relative service point locations and various loads. 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 increases the load on the servers, which may eventually decrease 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. In this paper, we present 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 of data and tries to match its chunks with a previously received chunk chain or a chunk chain of a local file. Using the long-term chunk's metadata information kept locally, the server receives predictions from receiver that include chunks signatures and easy-to-verify hints of the sender's future data. If the hint matches the sender first examines the hint and performs the TRE operation. 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 instead of sending data sends only the ACKs to the predictions to the receiver, instead of sending the data.

Pack: In this construct, we have a tendency to approach a unique receiver aspect primarily based destination to-destination TRE answer that depends on the ability of predictions to reject the redundant traffic between the cloud and additionally its users. In this formerly received chain using chunk's metadata If yes, then receiver sends a prediction starting point in the byte stream, the offset and the identity of several subsequent chunks(PRED command) Upon a successful prediction, sender responds PRED-ACK Then receiver copies the corresponding data from the chunk store to its TCP buffer, according to the corresponding sequence number Then sender respond by normal TCP ACK with the next expected TCP sequence number If prediction is false, Sender continues with normal operation. It tries to match it with its buffered data by receiving PRED message from receiver. Upon a hint match it (sender) calculates SHA-1 signature for the predicted range then compares the result to the signature received in the PRED message, PRED-ACK message is sent If hint not matched signature operation is not performed continuation are frequently interrupted, In this turn, forces the sender to retransmit to the raw data transmission until a new comparison is found at the receiver side and It reported back to the sender Side. To that end, we have to present the Predictive ACK hierarchal mode of operation.

II. PROPOSED WORK

Many Redundancy Elimination techniques have been explored in recent years. A protocol freelance Redundancy Elimination was planned in This paper was describes a sender packet-level Traffic_Redundancy Elimination, utilization of the rule given in several industrial Redundancy Elimination answers that delineate in and the protocol specific optimization technique for middle box solution have combined the sender based TRE concepts with the rule and implement approach of PACK. In necessary have to be compelled to describe the way to escape with this tripartite hand shake between the senders half and additionally the receiver half if any full state synchronize maintain. TRE system is applied for developing world wherever storage and WAN information measure are very rare. It's an application primarily based and connected middle-box replacement for the overpriced industrial hardware. During this kind, the sender middle-box holds back the TCP stream and sends data signatures to the receiver middle-box. The receiver checks and verifies whether or not the info is found in its native or primary cache. Information chunks that are not found in the cache are fetched from the sender middle-box or a near receiver middle-box. Naturally, for non cached information such a theme brings a three-way-handshake (3WH) latency.

A. TRE environment in cloud computing

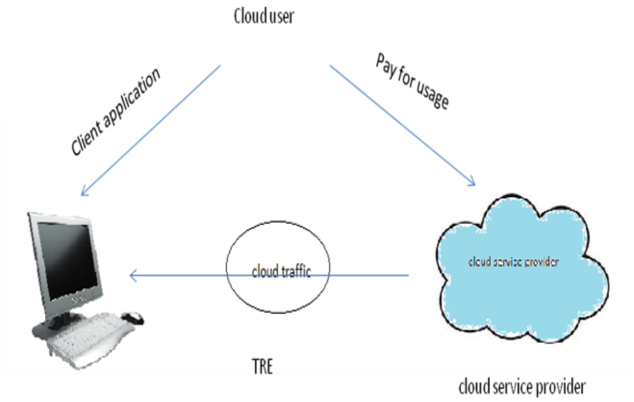


Fig: 1. Illustration of Cloud traffic in Traditional model

The above fig shows how the traffic encounters in the cloud computing between end user and the cloud server by transferring the same content repeatedly. In order to removing redundancy we are using TRE approach. Large amount of popular or the commonly requested content is transferred repeatedly across network links in the internet. To transfer the information between the sender and receiver data is divided into chunks. Chunking mechanism helps to improve the efficiency by parallel upload/download of different chunks which in turn generates a signature to transfer the data in a secure manner.

B. TRE based Wide Area Network

Traffic happens between the sender and receiver by transmitting the same information repeatedly, that information may contain files, documents, video's etc. In recent years various TRE techniques have been developed for eliminating the redundant (repeated) data. Wanax present, Wide-area network (WAN) accelerators operate by compressing redundant network traffic from point-to-point communications, which leads to higher effective bandwidth.

Multi Resolution Scheme:

Wanax uses a multi-resolution chunking (MRC) scheme that provides high compression rates and high disk performance for a variety of content, with the usage of much less memory

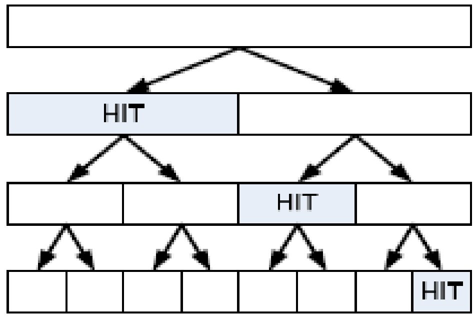


Fig: 2. MRC Aligned Chunk Boundaries

The above fig shows MRC (multi resolution chunking) operation; here the received data is divided into equal sized chunks. Content fingerprinting (CF) forms the basis for Wanax, since it provides a position-independent and history-independent technique for breaking a stream of data into smaller pieces/ chunks, based only on their information in it.

C. Significance of PACK approach

The stream of data received at the PACK receiver is parsed to a sequence of variable-size, content-based signed chunks. 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. Using the constructed chain, the receiver sends a prediction to the sender for the subsequent data. Part of each chunk's prediction, termed a hint, is an easy-to-compute function with a small-enough false-positive value, such as the value of the last byte in the predicted data or a byte-wide XOR checksum of all or selected bytes.

The prediction sent by the receiver includes the range of the predicted data, the hint, and the signature of the chunk. The sender identifies the predicted range in its buffered data and verifies the hint for that range. If the result matches the received hint, it continues to perform the more computationally intensive SHA-1 signature operation. Upon a signature match, the sender sends a confirmation message to the receiver, enabling it to copy the matched data from its local storage.

D. 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)
- Identity of several subsequent chunks (PRED command).
- Upon a successful prediction, the sender responds with a PRED-ACK confirmation message.

Once the PRED-ACK message is received and processed, the receiver copies the corresponding data from the chunk store to its TCP input buffers, placing it according to the corresponding sequence numbers. At this point, the receiver sends a normal TCP ACK with the next expected TCP sequence number. In case the prediction is false, or one or more predicted chunks are already sent, the sender continues with normal operation, e.g., sending the raw data, without sending a PRED-ACK message.

Steps for Receiver Segment Processing in PACK

1. Get the cloud user request from CSP
2. Analyze the service availability by comparing in data centers.
3. Ensure the bandwidth capacity for the requested user.
4. Apply TRE mechanism to validate the cost and bandwidth in cloud environment.
5. Use the PACK input segments to validate the result set
6. Responses can be obtained from middle box viewer t to process the application.
7. Direct these input and output consoles and target result to CSP.

E. Algorithmic Computation of Receiver segment values in PACK

1. **if data** segment carries payload *data* **then**
2. Calculate chunk for service bandwidth
3. **if** reached chunk boundary service limit **then**
4. Activate predAttempt () for cloud service access range
5. **end if service1 ()**
6. **else if** PRED-ACK segment **then**
7. Process PredAck() for requested service
8. Activate predAttempt()
9. **end if service n ()**

F. Flow chart Representation for pack algorithm

In the below figure receiver request the data from the cloud (sender). The data will arrive from the sender and the data is divided in the form of chunks, each chunk is assigned with a unique signature. The signature of the chunks is compared in the store if the chucks are matched receiver sends a predictive acknowledgement, if not continue with the normal operation. The Pack can be represented in the following diagram. It is mainly developed and constructed for reducing the internet bandwidth cost-cutting systems.

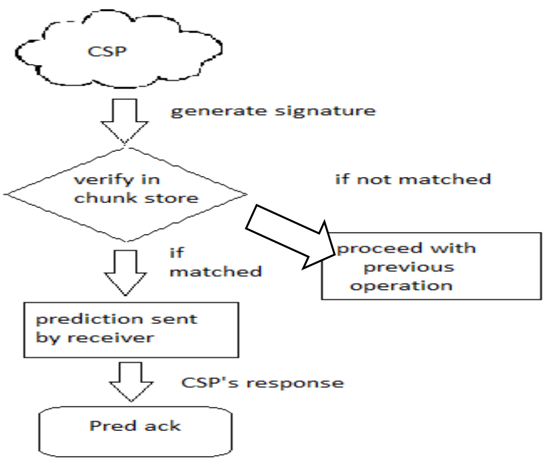


Fig: 3. Depicting PACK Algorithm

In the sender algorithm, the sender receives predictive acknowledgement from the receiver and checks it with buffered data. If the signature matches with the data sender calculates SHA-1 for the buffered data, if not matched signature operation is not performed. Check whether the generated signature is matched with the previous received signature. If matched send predictive acknowledgement to the receiver.

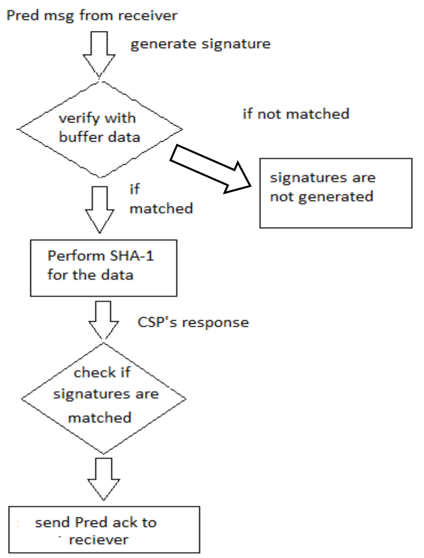


Fig: 4. Illustration of Sender algorithm

G. *End-Redundancy Elimination TRE*

End TRE end-system redundancy elimination provides fast, adaptive and parsimonious in memory usage in order to opportunistically leverage resources on end hosts. End TRE is based on two modules server and the client. The server-side module is responsible for identifying redundancy in network data by comparing against a cache of prior data and encoding the redundant data with shorter meta-data. The client-side module consists of following components:

- ➢ Fixed-size circular FIFO log of packets and simple logic to decode the meta-data by "de-referencing" the offsets sent by the server.
- ➢ Thus, most of the complexity in End TRE is mainly on the server side.
- ➢ Therefore the server specifically cannot maintain the full synchronization between client and the server.
- ➢ End TRE uses Sample Byte fingerprinting scheme which is faster than compared to Rabin fingerprinting.
- ➢ End TRE limited for small redundant chunks of the order of 32-64 bytes.
- ➢ Only unique chunks are transmitted between file servers and clients, resulting in the consumption of lower bandwidth. The basic idea underlying End TRE is that of ***content-based naming*** where an object is divided into chunks and indexed by computing hashes over chunks.

Comparison with Novel-TRE:

1. It is server specific
2. Chunk size is small.

It describes how to get aside with three-way handshake between the sender and the receiver if a full state synchronization is maintained. A method is used for reduction of network traffic. At a sender, a data chunk is identified for transmission to a receiver, which is connected to the sender over a communication link. The sender computes a signature of the data chunk and determines whether the date chunk has been previously transmitted by looking up the signature in a sender index table. The sender index table associates the signatures of previously transmitted data chunks with unique index values.

A message is transmitted to the receiver, where if the data chunk has previously been transmitted then the message includes an index value from the sender index table that is associated with the signature of the data chunk with the unique index values. At the receiver, the data chunk is located in a receiver cache that stores the previously transmitted data chunks by looking up the index value included in the message in a receiver index table. The receiver index table associates the unique index values with the locations in the receiver cache of the previously transmitted data chunks.

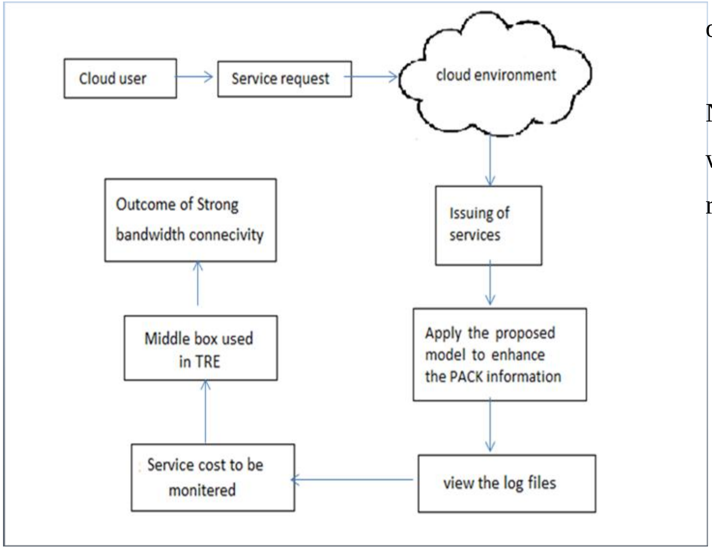
Packet level redundant content elimination as a universal primitive on all internet routers, such a universal deployment would immediately reduce link loads everywhere. However, we argue that far more significant network-wide benefits can be achieved by redesigning network routing protocols to ability to get high returns with the universal deployment.

The "redundancy-aware" intra- and inter-domain routing algorithms show that they enable better traffic engineering, reduce link usage costs, and enhance ISPs' responsiveness to traffic variations. Disadvantage Of course, deploying redundancy elimination mechanisms on multiple network routers is likely to be expensive to start with. However, we believe that the significant long term benefits of our approaches offer great incentives for networks to adopt them.

III. SYSTEM ARCHITECTURE

In this model cloud user sends a service request to cloud service provider (CSP). While issuing the services it applies the proposed model to enhance the PACK information .it views the log files and monitors the service cost. In this process middle box is used in TRE (Traffic Redundancy Elimination).The output of this model is strong bandwidth connectivity

5. Illustration of System Framework.



Significance of Novel TRE

The novel-TRE approach relies on the power of predictions to eliminate redundant traffic between its end-users and the cloud. In this technique, each receiver observes the incoming stream and tries to match its chunks with a previously received chunk chain or a chunk chain of a local file. Using the long-term chunks' metadata 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. On the receiver side, we propose a new computationally lightweight chunking [1] (fingerprinting) scheme.

Lightweight chunking is alternative for Rabin finger printing traditionally used by TRE applications with high data processing speed. The stream of data received at the novel-TRE receiver is parsed to a sequence of variable-size, content-based signed chunks. 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.

Using the constructed chain, the receiver sends a prediction to the sender for the subsequent data. Part of each chunk's prediction, termed a *hint*, is an easy-to-compute function with a small-enough false-positive value, such as the value of the last byte in the predicted data or a byte-wide XOR checksum of all or selected bytes.

H. Chunking Mechanism:

Novel-TRE uses a new *chains* scheme, described in Fig. 3, in which chunks are linked to other chunks according to their last received order.

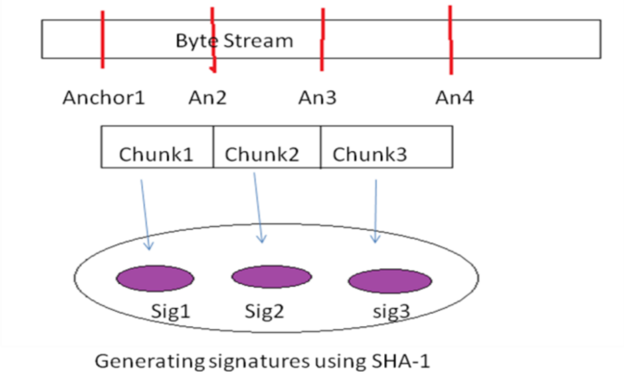


Fig: 6. Lightweight Chunking

The novel-TRE 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. When the new data are received and parsed to chunks, the receiver computes each chunk's signature using SHA-1. At this point, the chunk and its signature are added to the chunk store. In addition, the metadata of the previously received chunk in the same stream is updated to point to the current chunk..

I. Prediction Operation:

The Following figure shows how the chunks are predicting in the receiver, 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.

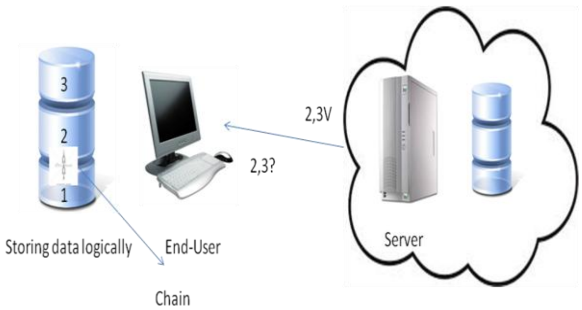


Fig: 7. Prediction Operation

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. Upon a successful prediction, the sender responds with a PRED-ACK confirmation message. Once the PRED-ACK message is received and processed, the receiver copies the corresponding data from the chunk store to its TCP input buffers, placing it according to the corresponding sequence numbers.

At this point, the receiver sends a normal TCP ACK with the next expected TCP sequence number. In case the prediction is false, or one or more predicted chunks are already sent, the sender continues with normal operation, e.g., sending the raw data, without sending a PRED-ACK message.

On the other hand, the use of smaller chunks increases the storage index size, memory usage, and magnetic disk seeks. It also increases the transmission overhead of the virtual data exchanged between the client and the server.

IV. IMPLEMENTATION WORK

The PACK implementation, its Performance analysis, and the projected server costs derived from the implementation experiments.

PACK receiver -sender protocol is embedded in TCP option Field Low overhead and compatibility with legacy systems Keep genuine OS's TCP and protocol above TCP Chunking and indexing at the client side only Decide independently on their preferred chunk size To achieve high TRE hit-ratio, while only negligible overhead of 0.1% Parameters Server operational cost PACK impact on the client CPU Chunking scheme PACK message format.

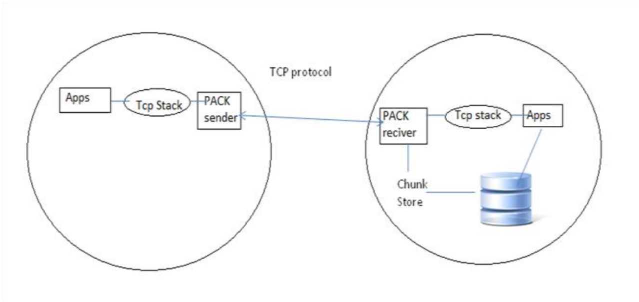


Fig 8: Overview of the PACK implementation

An implementation work enables the transparent use of the TRE at both the server and the client. PACK receiver–sender protocol is embedded in the TCP Options field for low overhead and compatibility with legacy systems along the path. We keep the genuine operating systems' TCP stacks intact, allowing a seamless integration with all applications and protocols above TCP.

V. SIMULATED RESULTSET

In cloud environment, without any interference cloud user need to get the service. The proposed approach mainly focuses on traffic elimination and service cost-cutting systems. By making use of any cloud resources; an end user needs to have strong bandwidth connectivity. For the simulation work, the following parameters are considered:

- User service request
- Access path Location
- TRE service portal
- PACK information set

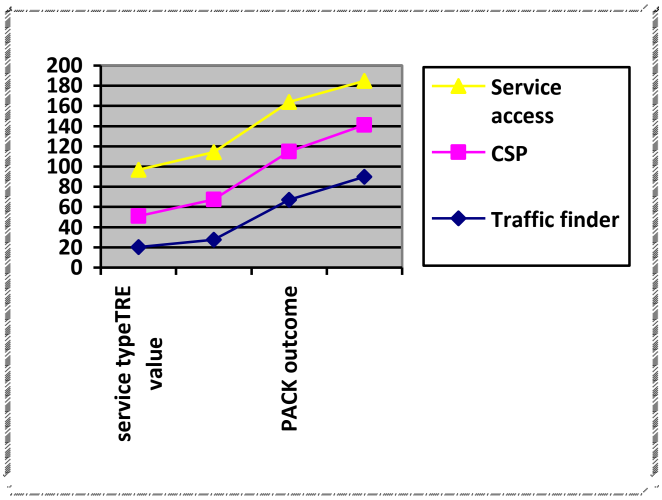
The following result set shows the targeted values in cloud environment.

User Service Request	Access path Location	TRE service outcome (TRE %10)	PACK based service efficiency (% 100)
SAAS	192.168.10.251	8.021	79.81
PAAS	192.168.109.62	6.691	80.62
DAAS	193.184.10.32	9.061	88.02
CAAS	192.168.10.265	9.24	90.92
NAAS	192.168.20.542	8.021	91.56
IAAS	191.169.10. 26	8.058	90.03

An above result set has been obtained, by simulated in the Java Net beans IDE environment. The input values are processed in green cloud simulator tool.

VI. EXPERIMENTAL RESULT

TRE mechanism was applied in the proposed model to eliminate the traffic problem in cloud environment. By the use this model, users can make use any service in cloud platform with PACK Technique.



VII. CONCLUSION

Increasing number of users in cloud computing are expecting high demand for TRE solutions in the amount of data exchange. TRE system requirements are redefined in the cloud environment which results in inadequate middle-box solutions. So there is a need for such TRE solution that reduces cloud operational cost.

PACK a receiver based end-to-end TRE Reduce latency and cloud operational cost doesn't require the server to continuously maintain client's status, thus enabling cloud elasticity and user mobility removing redundancy without applying a three-way handshake.

Evaluation using wide collection of content types Implementation maintains chains by keeping for any chunk only the last observed subsequent chunk in LRU.

VIII. FUTURE ENHANCEMENT

The proposed model has simulated in cloud environment with its input parameter in TRE mechanism. An outcome of these approach users can make use of any service in a continuous manner. Also it will direct to the future uses such as business applications, enterprise resource planning and other activities can be enabled.

REFERENCES

[1]. Author: chiwande V.N, Tayal A.R paper entitled as, "**An Approach to Balance the Load with Security for Distributed File System in Cloud" –** ICESC 2014.

[2]. Author: Shen Chen, Wenyu dong hang Li, peng Zhang, junwei cao paper entitled as, **"Collaborative network security in multi-tenant data center in cloud computing"** IEEE transaction volume-17, number 1, February 2014.

[3]. Author: Mendes. L.D.P, Rodrigues, Lloret.J, Sendra.S Paper entitled as**, "Cross – layer dynamic admission control for cloud- based multimedia senor networks"** IEEE transactions , 2014**.**

[4]. Author: Paper entitled as**, "A Policy-Based Security Framework for Storage and Computation on Enterprise Data in the Cloud"** IEEE transactions volume -4 march 2014**.**

[5]. Author: vijeng. L**,** Paper entitled as, **"Security in the cloud based systems: Structure and breaches "Internet Technology and Secured Transactions (ICITST) 2013.**

[6]. Author: Sheng Di, Member, IEEE, Cho-Li Wang, Member, IEEE, and Franck Cappello, IEEE Member, paper entitled as, "**Adaptive Algorithm for Minimizing Cloud Task Length with Prediction Errors**" IEEE transactions on cloud computing, vol. 2, no. 2, April-June 2014.

[7]. Author: Sudip Misra, Senior Member, IEEE, Snigdha Das, Manas Khatua, Student Member, IEEE, and Mohammad S. Obaidat, IEEE Fellow, paper entitled as, "**QoS-Guaranteed Bandwidth Shifting and Redistribution in Mobile Cloud Environment**" IEEE transactions on cloud computing, vol. 2, no. 2, April-June 2014

[8]. Author: Amir Vahid Dastjerdi, Member, IEEE and Rajkumar Buyya, Senior Member, paper entitled as, "**Compatibility-Aware Cloud Service Composition under Fuzzy Preferences of Users**" IEEE transactions on cloud computing, vol. 2, no. 1, January-March 2014.

[9]. Author: Hossein Morshedlou and Mohammad Reza Meybodi paper entitled as, "**Decreasing Impact of SLA Violations: A Proactive Resource Allocation Approach for Cloud Computing environments**" IEEE transactions on cloud computing, vol 2, no.2, April – June 2014.

[10]. Author: Luca Ferretti, Fabio Pierazzi, Michele Colajanni, and Mirco Marchetti paper entitled as, "**Performance and Cost Evaluation of an Adaptive Encryption Architecture for cloud databases**" IEEE transactions on cloud computing, vol 2, no.2, April – June 2014.

[11]. Author: Xiaomin Zhu, Member, IEEE, Laurence T. Yang, Member, IEEE, Huangke Chen, Ji Wang, Shu Yin, Member, IEEE, and Xiaocheng Liu paper entitled as, **" Real-Time Tasks oriented Energy – Aware scheduling in virtualized clouds**" IEEE transactions on cloud computing, Vol.2, no.2, april-june 2014.

[12]. Author: Chun-Wei Tsai, Wei-Cheng Huang, Meng-Hsiu Chiang, Ming-Chao Chiang, and Chu-Sing Yang paper entitled as, "**A Hyper-Heuristic Scheduling Algorithm for Cloud**" IEEE transactions on cloud computing, vol. 2, no. 2, April-June 2014.