PREDICTION BASED CLOUD BANDWIDTHAND COSTREDUCTION SYSTEMOFCLOUD COMPUTING

A.Rasik and S. Raj Anand

VelTechMultiTech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai

ABSTRACT

In this paper, we present PACK (Predictive ACKs), a novel destination-to-destination traffic redundancy elimination (TRE) system, Itespecially designed for cloud computing customers. PACK's main advantage is its capability of offloading the cloud-server TRE effort to end clients, thus reducing the processing costs induced by the TRE algorithm. Not like earlier solutions, PACK is based on a novel TRE technique, which allows the client to use newly received chunks to identify previously received chunk chains, which in turn can be used as reliable predictors to future transmitted chunks.PACK need not require the server to continuously maintain clients' status. This makes PACK very comfortable for pervasive computation network environments that combine client mobility and server migration to maintain cloud elasticity.Cloud-related TRE needs to apply a judicious use of cloud resources so that the bandwidth cost reduction combined with the extra cost of TRE computation and data storage would be optimized. We present a new fully functional PACK implementation, transparent to all TCP protocol based applications and all network devices. Finally, we analyze and implement PACK benefits for cloud users, using traffic traces from various sources.

KEYWORDS

Predictive Acknowledgement,Traffic Redundancy Elimination System,Caching,Cloud Computing, Network Optimization.

I .Introduction

Cloud system offers their client to an economical and convenient pay of service model,that is also as usage based pricing.[2] Cloud clients(customers) pay only for the actual use of cloud resources[1], storage of data, and band-width of usage,so according to their changing needs, utilizing the cloud scalable and spring computational capabilities. In this, data transmission costs (that is bandwidth) its most important issue when trying to reduce the costs consequently[2], cloud beneficiary, applying a judicious use ofthe cloud's resources,and they are motivated to use lot of various traffic reduction techniques, in particular traffic redundancy elimination (TRE), for reducing bandwidth costs.Traffic redundancy systems is from natural customer activities, such as repeatedly accessing, downword, upword (i.e., backup),distribute, and modify the same or different information items Traffic Redundancy Elimination System is used to eliminate the transmit of redundant content of data and information, mean while it significantlyreduce the cost of network. In most common scenarios Traffic redundancy solutions both the sender side and the receiver side examine and match the signatures of all data chunks[3-5], parsed according to the information content, prior to their transmission. Ifany redundant or duplicate chunks are founded means, the sender replaces the transmission of all each redundant chunk with its powerful signature.Normal TRE solutions are popular at this enterprise level networks, and it involve the deployment of three or more protocol, and it state synchronized middle-boxes at two level that is the intranet entry points of data centers and also the branch offices,removing repetitive traffic in between them.

II .Related Work

Many Redundancy Elimination techniqueshas been explored in recent years. A protocol independent Redundancy Elimination was proposed in This paper was describes a sender packet-level Traffic Redundancy Elimination, utilization of the algorithm presented in many commercial Redundancy Elimination solutions that described in and have combined the sender-based TRE ideas with the algorithm and implement approach of PACK and along with the protocol specific optimizations technique for middle-box solution.

In Important have to describe how to get away with this three-way hand shake between the sender part and also the receiver part if any full state synchronize is maintain.TRE system for the developing world where storage and WAN bandwidth are scarce. It is a application based and related middle-box replacement for the expensive commercial hardware. In this type, the sender middle-box holds back the TCP stream and sends data signatures to the receiver middle-box. The receiver verifies whether the information is found in its local cache. Data chunks that are not found in the cache are fetched from the sender middle-box or a nearby receiver middle-box. Naturally, such a scheme incurs a three-way-handshake(3WH) latency for non-cached data.

ProposedSystem of the PACK

In this concept, we approach a novel receiver side based destination-to-destination TRE solution that relies on the power of predictions to reject the redundant traffic between the cloud and also its users. In this study, every receiver observes the receiving stream and tries to compare its chunks with a previously received chunk chain or a chunk chain of a temporary local file. Using the longterm information chunks and metadata information that keep locally,nowthe receiver sends to the server predictions that include chunks' signatures and very easy-to-verify hints of the sender's future data. On the incoming side, we invent a new traditionally low weight chunking scheme termed Predictive ACK chunking. Predictive Acknowledgement chunking is a future enhanced alternative for Rabin fingerprinting used by RE applications.

III. Advanced Pack Algorithm Concept

A) Receiver Side Chunk Storage

Predictive ACK uses the new continuous chains scheme that described in Fig. 1, in that everychunks are related to all other chunks by their recent received way of order. The Predictive ACK receivers have to keep a chunk storage, which it's a very large size cache of chunks and their metadata. Chunk'smetadata includes the data chunk's signature and a single pointer to the successive chunk in the recent received stream that contain this chunk. Cache and index technique are employed toefficiently maintain and retrieve the every stored chunks and its signature and the chains created by traverse the chunk pointers.

B) Receiver Side Algorithm

The arrival of a new information, the receiver side system that respective signature for every chunk and see the match in its local(temporary) chunk storage. If the chunk's match is founded, the receiver side determines whether its a part of a formerly received chunk chain, using the chunks' metadata (data about data) otherwise If affirmative, the receiver send a prediction to the sender side for the several new expected chain chunks. Itcarries a beginning point in the byte stream that is offset and the identity of several subsequent chunks.

C) Sender Side Algorithm

Sender receives a Predictive message from the receiver side and it tries to compare the received predictions to its buffered yet to be sent Information. For every prediction, the sender have to determines that correspondingTCP range of sequence and verifies it. if that hint match, the sender measures the more computationally intensive Secure Hash Algorithm- 1 signature for the predicted information range and match the result tothe signature received in the Predictive message of data. In this case if the hint does not same, a computationally expansive operation is saved. If the two Secure Hash Algorith-1 signatures compare, the sender can safelyassume that the receiver's prediction method is absolutely correct. and, it replace the entire outgoing buffered data with a Predictive ACK message.

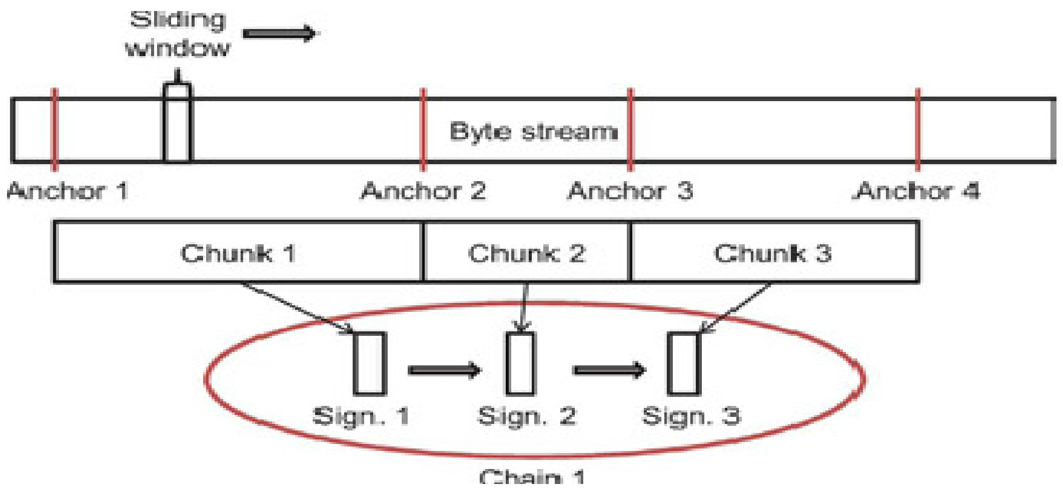


Figure 1: From the Stream Line to Chunk Chain

IV. Optimization

For the sake of purity, Part three presents the most vital basic versionof the Predictive ACK protocol. In this part, we have to describe the additionaloptions and optimization.

A)Adaptive Receiver Virtual Window

Predictive ACK enabling the receiver side to locally capture the sender data when a local or temporary copy is available, thus eliminating the requirement to send this information through the network. In this term the receiver's fetching of that recent local data as the reception of visual data.

B) Cloud Server Acting as a Receiver

In a developing trend, cloud computing storage is getting a dominant player from backup of store and sharing of data services to the American National Library and e -mail services . In this most of theseServices, the cloud is used often the receiver of the data.

C) Hierarchal Approach

Predictive ACK's receiver side based mode is less amount of efficient if changes in the information are scattered. In this scenario, the prediction 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 hierarchalmode of operation.

V.Motivating the Receiver Based approach.

The main objective of this part is to fold and Evaluate the potential data redundancy for most real time applications that are most likely to reside in a cloud, and to estimate the Predictive ACK performance and cloud network amountof the redundancy elimination process flow. Our process areconducted using.

1) Video traces captured at a major Internet Service Provider.
2) Trafficanalyses from the popular social networks and service.
3) Genuine data sets of real time works.

In this we related to an average size of chunk is 8 KB and although this algorithm allows each cloud client can use a different type of chunk size

VI. Estimated Cloud Cost for YouTube TrafficTraces

As noted earlier, although TRE reduces cloud traffic costs, the increased server computational efforts for TRE computation result in increased server-hoursi.e time cost.Without TRE, with PACK and with a sender-based TRE. The cost comparison takes into account server-hours and overall outgoing traffic throughput, while omitting storage costs that we found to be very similar in all the examined setups. The baseline for this comparison is our measurement of a single video server that outputs up to 350 Mbps to 600 concurrent clients. Given a cloud with an array of such servers, we set the cloud policy to add a server when there is less than 0.25 CPU computation power unemployed in the array. A server is removed from this array if, after its removal, there is at least 0.5 CPU power left unused.

VII. Evaluation

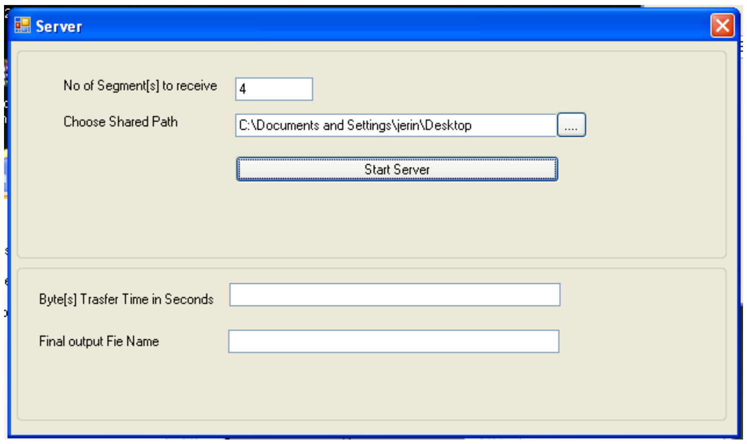
The objective of this section is twofold: evaluating the potential data redundancy for several applications that are likely to reside in a cloud, and to estimate the PACK performance and cloud costs of the redundancy elimination process. Our evaluations are conducted using (i) video traces captured at a major ISP, (ii) traffic obtained from a popular social network service, and (iii) genuine data sets of real-life workloads. In this section, we relate to an average chunk size of 8 KB, although our algorithm allows each client to use a different chunk size.

VII. Result and discussion

In this Implementation section, we present Predictive ACK implementation, its performance behavior analysis, and the projected server costs derived from the implementation experiments. Our implementation process contains nearly 25 000 lines of C and Java code. It works on Linux with Net filter Queue approach.The Predictiv ACK implementation architecture. At the server side, we use aIntel Core 2 Duo 3 GHz, 3 GB of Random Access Memory, and a SATAdrive desktop. The client'ssystem are based on an Intel Dual core 4.8 GHz, 4.5 Giga Byte of Random Access Memory, and a SATA drive.

A)Server Operational Cost

The server is required to perform an SHA-1 operation over a defined range of bytes only after it verifies that the hint or message, sent as a part of the prediction, compare the data. The sender side based TRE, the server is needed to first and foremost compute Rabin finger prints in order to piece the data stream into chunks, and then to compute an Secured Hash Algorithm-1 sign for all data chunk chain, prior to sending it.The server computational effort for the different sender side based and PACK TRE schemes, we evaluate the server effort as a function of time and function of cost and traffic redundancy. For the sender-based scheme, we simulated the approach of using their published performance benchmarks.



No of Segment[s] to receive: 4

Choose Shared Path: C:\Documents and Settings\jerin\Desktop

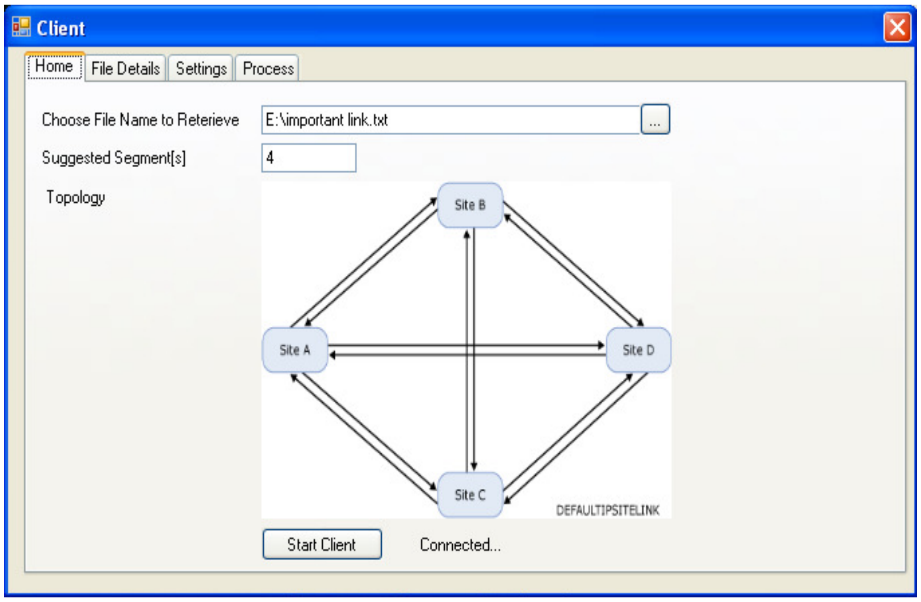
Byte[s] Trasfer Time in Seconds:

Final output Fie Name:

Figure2:server performance cost(SPC)

B)PACK Impact on the Client CPU

To review the CPU performance apparaisal that imposed by PACK on the client system, we measured that a random client under a specific condition similar to the one used for evaluating the server's performance cost, and only this time the cloud server streamed the videos at the rate of 9 Mega bytes to each and every client.Such a speed throttling is very common in real-time video servers that aim to provide all the clients with stable and standard bandwidth for smooth and clear view.


Client

Home | File Details | Settings | Process

Choose File Name to Reterieve: E:\important link.txt

Suggested Segment[s]: 4

Topology

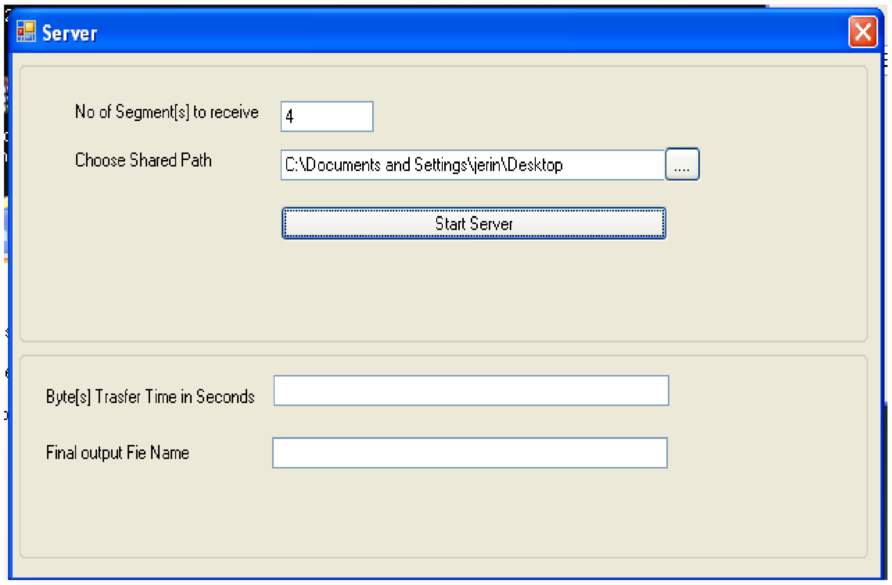
Figure 3:PACK Impact On The CPU

C)Pack Messages Format

The most important one is an enabling the option PACK permitted tosent in a SYN(synchronization) segment to indicate that the PACK option can be used after the connectivity link is established.Theanother one is a Predictive ACK message format that may sent over an established connection once a permission has been granted by both parties. This not only saves message overhead, but also copes with security network devices (e.g., firewall) that tend to change TCP protocol options order. Due to the lack of space and additional implementation details are left out and are available.

D)Server Computational Effort

The server is needed to perform an Secured Hash Algorith-1 operation over a defined range of bytes stream only after it verifies that the hint or message, sent as a specific part of the prediction, compares the data in the sender side based Traffic Redundancy Elimination, the server is required to first compute Rabin fingerprints in order to piece the data stream into chunks, and then to compute an Secure Hash Algorithm-1 signature for every chunk chain, prior to sending it. The server operational effort for the different sender-based and Predictive ACK TRE schemes, we evaluate the server effort as a function of time and traffic redundancy. For the sender-based scheme, we simulated the approach of using their published performance benchmarks.



No of Segment[s] to receive 4

Choose Shared Path C:\Documents and Settings\jerin\Desktop

Byte[s] Trasfer Time in Seconds

Final output Fie Name

Figure 4:Server Computational Effort(SEC)

System Avg.	Chunk Size Sender	Chunking Sender	Signing Receiver	Chunking	Receiver Signing
PACK	Unlimited, receiver's choice	None	SHA-1: true predictions and 0.4% of false predictions	PACK chunking: all real data	SHA-1: all real data
Wanax [12]	Multiple	Multi-Resolution Chunking (MRC):all data	SHA-1: all data	Multi-Resolution Chunking (MRC):all data	SHA-1: all real data
LBFS [20]	Flexible, 8KB	Rabin: all modified data (not relying on database integrity)	SHA-1: all modified data (not relying on database integrity)	Rabin: all modified data (not relying on database integrity)	SHA-1: all modified data (not relying on database integrity)
[3]	None (representative fingerprints)	Rabin: all data (for lookup)	(see Sender Chunking)	None	None
EndRE [1] Chunk-Match	Limited, 32-64 bytes	Sample Byte: all data(optionally less, at the cost of reduced compression)	SHA-1: all data (for chunk lookup)	None	None

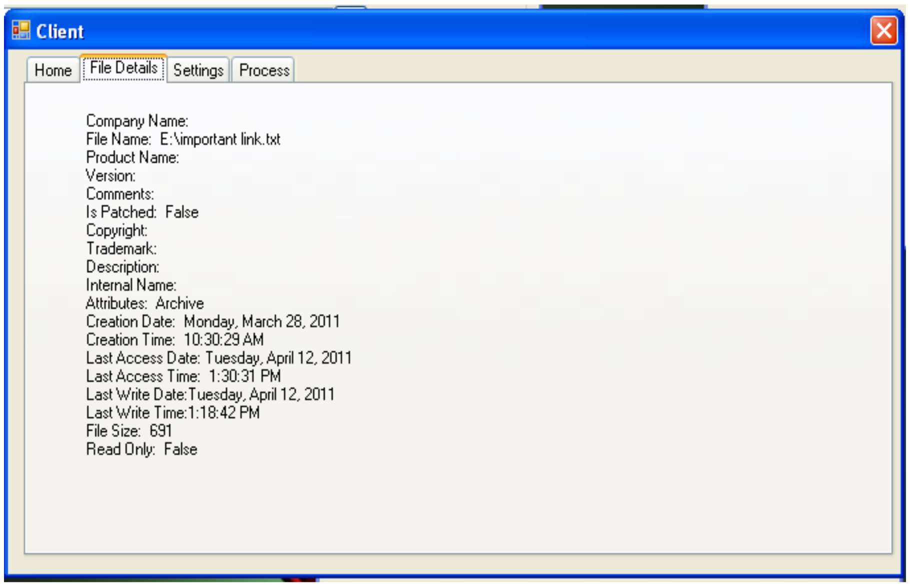
Table 1: Sender Computational Effort Comparison Between Different TRE Mechanisms

E)Synchronization

Many sender side based end to end Traffic Redundancy mechanisms require full synchronization between the sender and the receiver caches. Whensuch synchronization exists.The redundancy is detected and eliminated up front by the sender. In this synchronization saves an otherwise needed to three-way handshake(3WH), it eliminates the redundant chunks that arrive at the receiver side from to different senders. A short-term cache did not identify returning to long-term sessions.

F)User Mobility (Receiver Side)

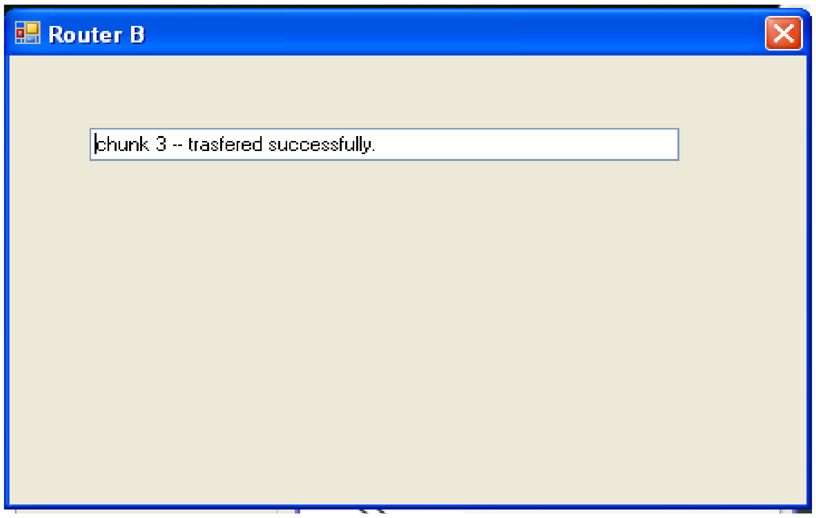
The common social network datasets are presented above, we explored the effective use of users' mobility on TRE. TRE solutions that are attached to a most specific location or rely on static client Internet protocol (IP) address cannot exploit this redundancy. This is what also a major problem for synchronized solutions that need a reliable protocol-independent detection of returning clients.



```
Company Name:
File Name:  E:\important link.txt
Product Name:
Version:
Comments:
Is Patched:  False
Copyright:
Trademark:
Description:
Internal Name:
Attributes:  Archive
Creation Date:  Monday, March 28, 2011
Creation Time:  10:30:29 AM
Last Access Date: Tuesday, April 12, 2011
Last Access Time:  1:30:31 PM
Last Write Date:Tuesday, April 12, 2011
Last Write Time:1:18:42 PM
File Size:  691
Read Only:  False
```

G)Chunking Scheme

Our output employs a novel computationally low weight chunking scheme, termed Predictive ACK chunking Tailored for the fast TRE chunking method. Anchors are detected by the mask in line 1that provides on average 8-kilo Bytes ofchunks.



chunk 3 -- trasfered successfully.

Figure: Chunking Scheme

VIII. Conclusion

Cloud computing is expected to trigger the high demand for TRE. In this Implementation the amount of data send between the cloud and its users is expected to diagrammatically high. The cloud computing environment redifines the TRE system and its requirements, making proprietary middle -box results inadequate. Consequently, there is a increasing need for a Traffic Redundancy Elimination solution that decreases the cloud's operational cost while accounting for application latency, user mobility, and also the cloud elasticity. In this paper, we have presented PACK, a receiver-based, cloud friendly, end - to-end TRE that is based on novel speculative principles that reduce latency and cloud operation cost.

PACK no need to require the server to permanently maintain clients' status, is enable the cloud elasticity and user mobility i.e. any time anywhere while preserving long term redundancy. Moreover, PACK have the strong capability of rejecting the redundancy based on content arriving to the client from different servers without applying the three-way handshake(3WH). Our evaluation using a large collection of material types shows that Predictive ACK meets the expected design goals and also have clear advantages over sender -based TRE, most especially when the cloud computation network cost and buffering requirements are most important. Moreover, PACK Implements additional effort on the sender side only if redundancy is exploited, then its consequently reducing the overall cloud cost**.**

References

[1] E. Zohar, I. Cidon, O. Mokryn,"The power of prediction: Cloud bandwidth and cost reduction", In Proc. SIGCOMM, pp. 86–97,2011.
[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski , G. Lee, D. Patterson, A. Rabkin, I. Stoica, M.Zaharia,"A view of cloud computing", Commun . ACM, Vol.53, No. 4, pp. 50–58, 2010.
[3] U. Manber,"Finding similar files in a large file system", in Proc. USENIX Winter Tech. Conf., pp. 1–100,1994.

[4] N. T. Spring, D. Wetherall,“A protocol-independent technique for eliminating redundant network traffic”, In Proc. SIGCOMM,Vol. 30, pp. 87–95,2000.
[5] A. Muthitacharoen, B. Chen, D. Mazières,“A low-bandwidth net-work file system”, In Proc. SOSP pp. 174–187,2001.
[6] E. Lev-Ran, I. Cidon, I. Z. Ben-Shaul,“Method and apparatus for reducing network traffic over low bandwidth links”, US Patent 7636767,Nov 2009.
[7] 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.
[8] R. Williams,“Method for partitioning a block of data into subblocks 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.
[11] http://en.wikipedia.org.
[12] http://www.PPStream.com.
[13] http://www.pplive.com.
[14] http://www.sopcast.com.
[15] http://www.tvants.com.