WAN OPTIMIZATION BY APPLICATION INDEPENDENT DATA REDUNDANCY ELIMINATION (BY CACHING)

Manjunath R Bhat[1], Bhishma[2], Kumar D[3], Kailas K M[4], Basavaraj Jakkali[5]

[1,2,3,4] Pursuing B.E in Computer Science & Engineering, BMSCE, Bangalore
[5]Associate Professor Dept. of CSE, BMSCE, Bangalore, India

Abstract: Huge amount of data flows through internet today, the data traffic over the internet is more than 25,000 GB per second. This leads to problems such as congestion, latency, low throughput etc. So for a smooth user experience the Wide Area Network (WAN) should be optimized. Many techniques called WAN optimization techniques are available in order to increase data-flow efficiencies across wan. Main wan optimization techniques include deduplication, caching, latency optimization, traffic shaping, protocol spoofing etc. This article mainly discusses the deduplication technique for wan optimization.

Deduplication eliminates the transfer of redundant data across the network, it does so by sending a reference to data and not sending the data itself. Deduplication is also called as Data Redundancy Elimination or DRE. DRE can be implemented either at packet level where we search and eliminate redundant packets or at byte level where we search for repeated byte string and replace it with a shorter reference. Surveys have shown that the DRE has ability to reduce the data traffic by 15% to 60%.

This article discusses major DRE techniques which are available and challenges for implementing DRE algorithms.

Keywords: Deduplication, caching, chunk matching, latency, fingerprinting, indexing

I. Introduction: The amount of data which flows through internet has become very large due to new technologies such as High Definition video streaming, VoIP, video calling etc.[1] But low power servers and underlying hardware cause lag and other inconvenience to users. So to increase user convenience and network efficiency we need WAN optimization [2]. Many techniques are available for optimizing the wide area network, some of them are

Compression: Reducing the size of the data that has to be transmitted by identifying patterns in the data [4].

Latency reduction: Different techniques used to reduce the time spent by the packet inside the network [5].

Caching: Caching the file requested by a user in the local server, if any other user requests, the same file then the locally stored file is served. This reduces the data that is flowing through network as well as the load on the main server [6].

Deduplication: Sending a reference instead of sending redundant data [7].

Traffic shaping: Controlling the data flow from specific applications or users [8].

This article discusses mainly about the Deduplication or Data Redundancy Elimination (DRE) technique.

Redundant data flow in network make the network slower, wastes network resources and decreases network performance. Due to increase in network traffic eliminating redundancy has become one of the key area of research in recent years. Redundancy elimination solutions mainly include eliminating redundancy at the object level and eliminating redundancy at packet level. Object level redundancy elimination works at application layer and it includes techniques such as data compression, web caching etc. Packet level redundancy elimination works at packet level and avoids sending redundant packets over the network. Packet level DRE techniques are usually protocol independent [7].

The basic technique in DRE is to store every outgoing and incoming data packets in cache and if the same packet has to be sent once again then only send a reference that points to the stored packet. Fig 1 represents this process.

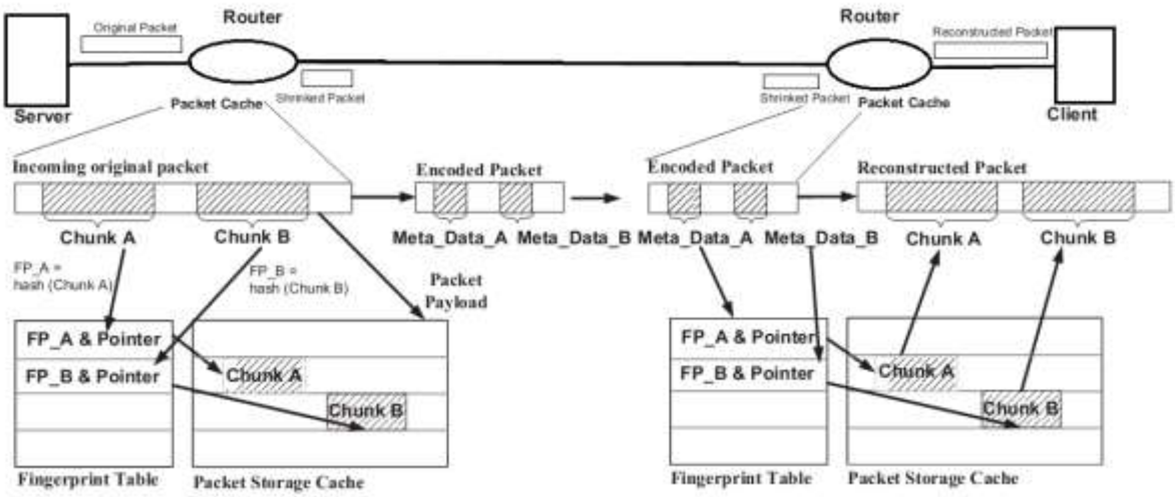

Fig 1. Basic technique for DRE.

Different surveys based on some terabytes of data have shown that by using DRE algorithms can reduce up to 60% total network traffic. Rest of the paper discusses about various implementations of DRE algorithms, main challenges involved in implementing these algorithms in real world and future works that can be done in this area. First we discuss main components of DRE like fingerprinting, cache management, and block matching. Then we present some of the implementations of DREs.

II. DRE (Caching): In DRE techniques every incoming packet is hashed using some hash function and a digest often called as ‘fingerprint’ of the packet is created. This digest or fingerprint uniquely identifies packet or data block. Each fingerprint will be indexed and stored in a table called ‘fingerprint table’. Whenever a packet or data block with redundant data is encountered the data will be replaced with a metadata which will be smaller than the original data but represents the original data. The metadata will later be used by the receiver to recreate the original data[39].

Main elements of DRE techniques are represented in fig 2.

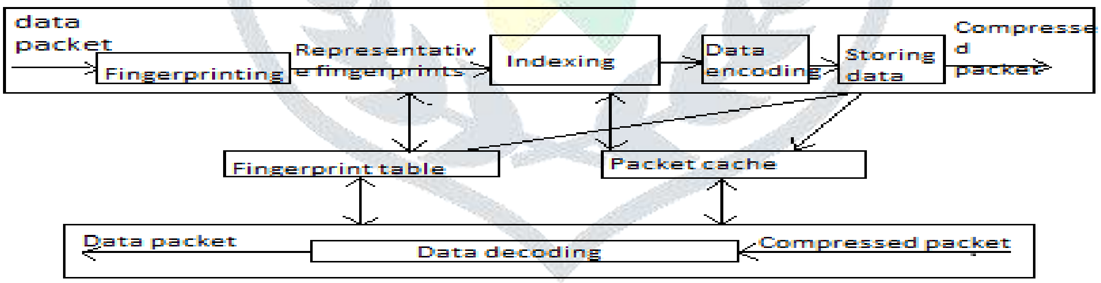

Fig 2. Main elements of DRE techniques

The main elements of DRE are fingerprinting, data matching and cache management.

A. Fingerprinting: In order to eliminate redundant data we need to identify redundant data blocks, since the fingerprint table is finite fingerprinting technique should be carefully selected to make the most of DRE algorithm [3].

There are several methods to do fingerprinting some of them are

1) Fixed fingerprinting

A fixed size n is chosen to select the data block for fingerprinting. The fingerprinting happens based on position of the data block. For every n byte of data a fingerprint is created.

This method is fast and very easy to implement but it's not that effective since it fails to identify similarity between two data blocks for very small modifications.

2) MODP fingerprinting

In this method a sliding window of size m is selected. For every data block of size m a fingerprint is created using Rabin-Karp hashing algorithm[17]. Since the fingerprint table is finite only 0 mod m th fingerprints are selected as constituting fingerprints.

This method is robust against minute modifications in documents but it is slow and generates very high overhead.

3) Winnowing [11]

In winnowing the fingerprints of data blocks are uniformly distributed. Winnowing uses a minimum and maximum threshold and creates fingerprints between that ranges. So for each segment there will be at least one fingerprint.

This method does uniform distribution of fingerprint but generates very high overhead.

4) Frequency based chunking [15]

The fingerprinting here is content based. Fingerprints are created for most frequent data blocks. This method uses a parallel filtering algorithm to calculate the frequency of data blocks.

Similarly there are many algorithms such as MAXP [12], sample byte [13], dynabyte [14,18] etc. from which we can chose one or hybrid of two algorithms based on our requirements.

B. Chunk Matching: There are mainly two chunk matching algorithms chunk match and max match [13]. In chunk match if a fingerprint matching the data is found in the fingerprint table then the data is marked as redundant data. In max match when a data block matches the fingerprint not only that chunk but the checking is extended to chunks left and right side until a miss match is found.

A new concept called combiheader[16] is becoming famous, it assumes that if two data blocks appears together then that two blocks will probably appear together in future. It keeps hit and miss counts for each block.

C. Cache management: Cache management includes discarding already cached packets and making space for new packets. Usually two mechanisms are used one is FIFO (first in first out) and LFU(least frequently used). A new policy called LSA (least savings with aging) is based on LFU but uses aging as a factor [4]. It keeps packets which can save more bandwidth if a hit occurs and discards the packets which does less bandwidth saving.

III. Challenges in implementing DRE algorithms:

A. Indexing: We identify each data block uniquely using fingerprints which are created using some hash function. These fingerprints are stored in a fingerprint table which should be indexed properly for fast retrieval of data from cache. The index should have mainly three important properties 1) should have very high capacity 2) indexing throughput should be more and 3) duplicate entries should be avoided.

Since the fingerprints are the output of hash functions, they are randomly distributed. If indexing takes up too much memory the searching time increases and if the memory used for indexing is less the redundancy may go undetected. So implementing indexing is a major problem in DRE.

B. Unique reference management: References used for identifying data blocks should be managed accurately. If a reference pointing to a deleted segment is used by any other files then the file is pointing to invalid or null data hence the file becomes corrupted or if a segment is referenced unnecessarily then it leads to memory leakage. References are also used to keep track of the memory usage information.

C. Synchronization on client and server: Since DRE sends references instead of actual redundant data the cache stored in the client side as well as the server side should be exactly same. Both systems cannot detect whether the same is data stored in other system also. So if one system sends a reference and the receiver cannot reconstruct same information based on the reference then the data cannot be retrieved. Implementing a mechanism which ensures that both systems are synced is very essential [9,10].

IV. Conclusion: In this paper we have reviewed the current state of DRE. We have presented a general structure of DRE solutions and its components. The important techniques used in DRE such as fingerprinting, cache management and data block management have been reviewed. And finally we discussed major challenges we might face while implementing DRE techniques in the real world.

DRE has proven to be more effective than other wan optimization techniques such as compression, web caching etc. but further research is needed in this area in order to effectively implement the solutions.

V. References:

[1] Williamson, C "Internet traffic measurement", in Internet Computing, IEEE (Volume:5 , Issue: 6), Nov/Dec 2001, pp. 70-74

[2] Zhang, Yan ; Ansari, Nirwan ; Wu, Mingquan ; Yu, Heather "On Wide Area Network Optimization", in Communications Surveys & Tutorials, IEEE (Volume:14 , Issue: 4), 13 October 2011, pp. 1090 – 1113

[3] Redundancy in network traffic: findings and implications "Redundancy in network traffic: findings and implications" in SIGMETRICS '09 Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems,Pages 37-48

[4] Welton, B, Kimpe, D. ; Cope, J. ; Patrick, C.M. "Improving I/O Forwarding Throughput with Data Compression" in Cluster Computing (CLUSTER), 2011 IEEE International Conference on, 26-30 Sept. 2011, pp 438-445

[5] Brian Trammell, Alessandro Finamore, Marco Mellia "A Measurement-Centered Approach to Latency Reduction", in Internet Society, November 2013

[6] Waleed Ali , Siti Mariyam Shamsuddin, and Abdul Samad Ismail "A Survey of Web Caching and Prefetching", in Int. J. Advance. Soft Comput. Appl., Vol. 3, No. 1, March 2011

[7] Yan Zhang, Nirwan Ansari "On Protocol-Independent Data Redundancy Elimination", in Communications Surveys & Tutorials, IEEE (Volume:16 , Issue: 1), 06 June 2013, pp. 455-472

[8] Bannazadeh, H "A Distributed Ethernet Traffic Shaping system", in Local and Metropolitan Area Networks (LANMAN), 2010 17th IEEE Workshop on, 5-7 May 2010, pp. 1 – 7

[9] Petros Efstathopoulos, Fanglu Guo, "Rethinking Deduplication Scalability", Symantec Research Labs, June 2010

[10] Fanglu Guo, Petros Efstathopoulos, "Building a High-performance Deduplication System", Symantec Research Labs

[11] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local Algorithms for Document Fingerprinting," in *Proc. 2003 ACM SIGMOD international conference on Management of Data (SIGMOD),* San Diego, California, 2003, pp. 76–85.

[12] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in Network Traffic: Findings and Implications," in *Proc. 11th International Joint Conference on Measurement and Modeling of Computer Systems*, Seattle, WA, USA, 2009, pp. 37–48.

[13] B. Aggarwal, A. Akella, A. Anand, A. Balachandran, P. Chitnis, C. Muthukrishnan, R. Ramjee, and G. Varghese, "EndRE: an End- System Redundancy Elimination Service for Enterprises," in *Proc. 7th USENIX conference on Networked systems design and implementation*, San Jose, California, 2010, pp. 28–28.

[14] E. Halepovic, C. Williamson, and M. Ghaderi, "DYNABYTE: A Dynamic Sampling Algorithm for Redundant Content Detection," in *Proc. 20th International Conference on Computer Communications and Networks (ICCCN)*, Maui, Hawaii, USA, July 31 - Aug. 4, 2011.

[15] G. Lu, Y. Jin, and D. Du, "Frequency Based Chunking for Data De-Duplication," in *Proc. Modeling, Analysis Simulation of Computer and Telecommunication Systems*, Aug. 2010, pp. 287 –296.

[16] S. Saha, A. Lukyanenko, and A. Yla-Jaaski, "Combiheader: Minimizing the number of shim headers in redundancy elimination systems," in *Proc. INFOCOM workshops*, April 2011, pp. 798 –803. [17] M. O. Rabin, "Fingerprinting by Random Polynomials," Center for Research in Computing Technology, Harvard University, Tech. Rep. TR-15-81, 1981.

[18] M. G. Emir Halepovic, Carey Williamson, "Low-Overhead Dynamic Sampling for Redundant Traffic Elimination," vol. 7, no. 1, Jan. 2012.

Author Profile:

Basavaraj Jakkali is an Associate Professor in the Department of Computer Science & Engineering at BMS College of Engineering. He has completed his Post Graduation from Vishweswaraya Technological University (VTU), Belgaum, in 2002. His areas of interest include Computer Organization, Microprocessors, Theory of Computation, Operating Systems, System Software and Network Security.

Manjunath R Bhat is Pursuing B.E in Computer Science & Engineering, BMSCE, Bangalore.

Bhisma is Pursuing B.E in Computer Science & Engineering, BMSCE, Bangalore.

Kailas K M is Pursuing B.E in Computer Science & Engineering, BMSCE, Bangalore.

Kumar D is Pursuing B.E in Computer Science & Engineering, BMSCE, Bangalore.