iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** I **Month of publication:** January 2018

DOI: http://doi.org/10.22214/ijraset.2018.1185

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

Survey of DELTA ++: An Efficient Encoding Technique for Android Application Updates

Syed Abdul Lateef Junaid[1], Afroze Ansari[2]
[1]*M.Tech CSE II year,* [2]*Assitant Professor, Dept of CSE, KBN College of Engineering, Gulbarga*

Abstract: ***Smartphone has more advanced computing capability and it uses android. Each day, Lakhs of android applications are being downloaded. These android applications are updated timely to remove errors if any and add new features. The key principle of developing a better technique to update mobile applications is to reduce the traffic generated by updates. But it is not sufficient to shrink the update size as much as possible. There is a negligible savings achieved by reduction of traffic because of the expensive changes, that require to be made in the software used to distribute application updates. Therefore, it is very significant to develop new updating methods that can be easily implemented on top of the existing infrastructure and with minimum modification in the software employed. In this paper, we review and compare the different DELTA encoding algorithms for reducing, the size of the application updates on android platform. Also we have carried out the review for the future research scope in delta ++ encoding.***

Keywords: samartphone, Android, Delta encoding, traffic, Smart, patch, DELTA++ encoding.

I. INTRODUCTION

Smartphone has more advanced computing capability as compare to other basic mobile phone. Smart phone is based on android and Linux kernel is developed by Google [1] [2]. In android market about 1.6 billion android applications available for Smartphones, as per survey done in the July 2015. These applications can have a number of bugs and there is a need to add new additional features in some cases.

There will be increase in traffic and load on data center wherever a new version of an application is downloaded whose old version is available in the smartphone entire application gets downloaded. The updates size can be decreased using delta encoding method and Instead of downloading the entire android application again, only the difference between the new and old versions can be downloaded on the smartphone.

Delta Encoding Method is used to develop google smart application update by google[3]. Delta encoding algorithms computes a difference among the target file and reference file.

The target file is constructed from the reference file by using patch(output). Delta Encoding method[4] can be used in cross platform applications for reducing the size of application updates and thus reducing network traffic in Blackberry, iOS and as well as in other mobile operating systems.

The major drawbacks of Smart Application Update are: Smart Application Update is not optimal, Data traffic is created instead of reducing it and the encoding method of Smart Application Update is Android Application Package (APK) level only. DELTA++ encoding algorithm will overcome all the drawbacks of Smart Application Update. DELTA++ is a method used to dispatch the difference between new and old versions of same android application. DELTA++ unpacks android application package (AAP) and compresses each element, thus there is a smaller patch to download [5].

The advantages of delta encoding are

1) DELTA algorithm decreases the size of application update

2) It also reduces the network traffic

DELTA++ is the improved version of DELTA which enables larger decrease in network traffic. Difference or diff is calculated by DELTA++ and Google Smart Application Update. Diff is the difference between older version and the newer version files. The diff is also known as patch. The content provider will be able to update a smart phone application by transferring the small patch (diff).The diff is applied locally on the smart phone. But DELTA++ unpacks the Android Application Package. DELTA++ compresses their individual module, which was not done in Google Smart Application Update. DELTA++ is more optimal, cost efficient and reduced network traffic to a greater extent as compared to other techniques [6].

II. LITERATURE REVIEW OF DELTA ENCODING ALGORITHM

Delta encoding algorithms aim to calculate a difference between the target and reference file, and efficiently encode this difference so that the target file is constructed from the reference file, by using the resulting patch. Levenshtein in 1965 first studied the file differencing problem that arises from the string-to-string correction problem [7].

The delta differencing algorithm was first proposed and implemented by J. W. Hunt and M. D. McIlroy in 1976 at bell lab in UNIX diff tool [8], which was used to efficiently store multiple versions of text documents.

In 1977 J. the W. Hunt and T. G. Szymanski proposed a fast algorithm for computing longest common sequences in two text files on a line-by-line basis. For better compression rates delta encoding algorithms uses techniques similar to the Ziv-Lempel compression algorithm [9].

In 1986 [10] Bdiff was developed and a modified Tichy's block move algorithm [11] is used, which works with file's blocks instead of lines, and thus can be efficiently used not only for text files. Bdiff can be considered a delta compression algorithm. In 1988 a splay tree algorithm is developed for data compression [12]. In 1996 a Vdelta algorithm is invented and it is a modification of the Tichy's block-move algorithm and it uses a greedy algorithm to match file blocks. The greedy algorithm decreases vdelta's running time and also memory usage, but it diminish its compression rate. Vdelta is the basis for the vcdiff delta encoding format [13]. Similar to vdelta's approach was used in zdelta [14], which is built up on the compression library called zlib [15], and in xdelta presented as a part of XDFS file system [16]. The zdelta and xdelta are based on vcdiff format. gdiff is widely used delta encoding format [17].

Baker et al. in 1999 [18] proposed that the size of the delta differences calculated between binary files can be significantly reduced by considering platform-dependent structure of executables.

Also the authors presented the exediff algorithm that has two main steps: preliminary matching and another is value recovery. The bsdiff algorithm is developed by C. Percival [19]. The bzip2 algorithm presented by J. Seward [20] is used to compress, the constructed patch, which in size is somewhat greater than the size of the patches, generated by platform dependent exediff.

A. DELTA: Delta Encoding for Less Traffic for Apps

Delta encoding method has been effectively used to reduce the amount of mobile network traffic in various systems such as internet browsing, distribution of software update and file systems.

Housel et al. [21] proposed to use delta encoding method called form differencing – in their WebExpress system. It employs a client/intercept model and does not need any changes in server or client. Two components or interceptors are installed: one on the client's side and another on the server's side.

Mogul et al. presented the advantage of using delta encoding in HTTP communications system [22]. This author has analyzed traces from real world web sites and used diff, vdelta, and gzip to reduce traffic for apps. Mogul et al. [23] that proposes RFC 3229 a compatible extension to HTTP/1.1 protocol that permits the server to send delta coded responses, if the client supports delta encoding algorithms. Such work is divided into 3 categories:

1) Software Updating
2) HTTP Communications
3) File Storage and Replication

Presently, many of the popular software is updated using delta encoding method that fits logically in the updating process as the main aim of it is to replace the old version with the new version. Microsoft created the Binary Delta Compression (BDC) method in 1998 with Windows NT 4.0 Service [24] and also the Delta Compression API [25] to distribute compact updates for its various windows operating systems.

B. Delta++: Reducing the Size of Android Application Updates

Google Smart Application Update has been developed by Google to decrease the size of the Android application update. It was used to construct a patch of the Android application installed on user's device. This is not an optimal solution to reduce the size of the android application update. Delta encoding method works on the Android Application Package level only. Delta encoding limits the possible reduction of the android patch. DELTA++ encoding method reduces network traffic more and attains greater savings as compare to delta encoding method. DELTA++ encoding computes the difference among the two files. It builds a patch, a newer version from the older version. Content provider is able to update smartphone application through transferring the difference between the older and newer version and then delta patch is applied locally on the smart mobile phone. DELTA++ unpacks the

Android Application Package (APK) and it will compress each module. The reduction of size of update by DELTA++ is 77%. DELTA++ utilizes more time to deploy as it needs an additional complex application patch. [5] [6].

Use of DELTA++ can give much more savings in the App Store than in Google Play because no method is at present used in the App Store to reduce android application updates traffic at the same time as Google Smart Application Update is employed in Google Play. The use of DELTA++ for Android and iPhone application updates can reduce yearly network traffic in cellular networks by more than 11.5% or by 140 PB. So there is significant savings for mobile operators and data centers as less network bandwidth and less number of servers, possible needed for serving patches[6].

Encoding algorithm	Advantages	Disadvantages
DELTA Encoding	i) A way of reducing traffic for smart phone app updates is developed and evaluated. Instead of full new version of the app only the difference among app versions is transferred. iii) The average size reduction of app updates, (for the top 50 most popular Android apps) was 48 48% reduction in app updates network traffic in carrier networks and also within data centers.	i) In delta encoding no server to host Android Apps and also no patches configuration. ii) Delta encoding limits the possible reduction in size of patch because it is at the Android Application Package (APK) level only,
Google Smart Application Update	i) load induce by online app updating will be reduce by this method ii) In this, battery life of mobile devices is increased iii) Savings in cellular networks can be enabled. Reduction in size of application will be 55%	i) Google Smart Application (GSA) carries out only a single clean update process. It is capable to collect any statistics collection, and accounting. ii) the compression method of this is not optimal. Iii) It creates, network data Traffic. iv) it had only the Android Application Package level encoding
DELTA++ Encoding	i) This method reduces the update network traffic. ii) This method is more optimal as compared to other encoding techniques. iii) this is Cost efficient iv) In this case decompression of Android Application Package and compression is done on every modules of APK. This is the key difference between DELTA++ and Google Smart Application Update	i) In this case no server to host Android Apps and also patches configuration. ii) As compare to other method this has optimal reduction but there will be mote reduction.

Table I: the comparison of different encoding algorithms

III. FUTURE RESEARCH SCOPE IN DELTA ENCODING: DELTA AND DELTA++

A. DELTA encoding based technique need to be further developed to update App Store applications.

B. Further there is scope for analysis and study of saving of network bandwidth and number of servers requirement of data centers, that serve application updates, if these updates can be distributed from one Smartphone to another Smartphone by use of P2P protocol. The application can be designed and employed to upgrade apps without downloading them from a data center, once a sufficient number of devices received an update.

C. Further, there is a need to study similarities between different applications and developing a technique to eliminate redundancy among them. This permits to reduce the size of android application update further.

D. There is a need for design of framework that includes all the designed methods and will gives users an easy way to efficiently update all the applications installed on their mobiles devices.

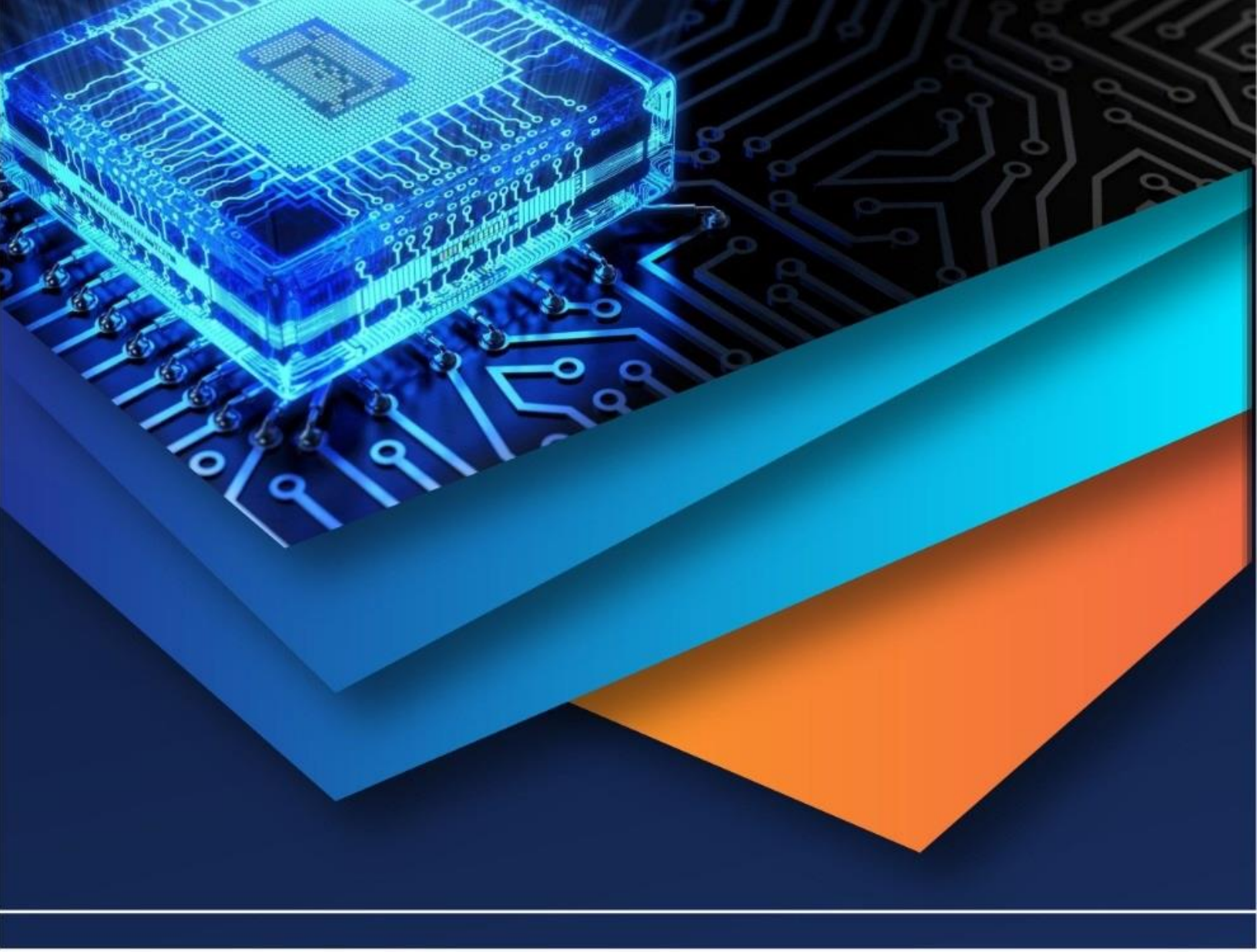
E. This framework can contain further optimizations of pre-fetching of the existing updates through Wi-Fi network whenever possible and then immediately updating applications through request of user. Availability of various updating techniques will permit users to specify different policies and make a decision whether they want to reduce the mobile network traffic or update application as quick as possible.

IV. CONCLUSIONS

We have reviewed and compare the DELTA encoding and Delta ++ algorithms for reducing the size of the application updates on android platform. We have analysed that for smartphones users, DELTA++ encoding can provide significant network traffic reduction and cost savings, if 50 sec delay can be accepted, during android application updating. Also we have identified the future research scope of Delta ++ encoding.

REFERENCES

[1] Nikolai Samteladze University of South Florida, nikolay.samteladze@gmail.com Delta Encoding Based Methods to Reduce the Size of Smartphone Application Updates http://scholarcommons.usf.edu/etd/4573
[2] S. Musil, “Google Play Enables Smart App Updates, Conserving Batteries”, CNET News, 16 Aug: 2012; http: //news.cnet.com=8301-10233-57495096-93/google-playenables-smart-app-updates-conserving- batteries/.
[3] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula and D. Estrin, A first look at traffic on smart phones, Proceedings of the 10th Annual Conference on Internet Measurement, pp. 281- 287, 2010.
[4] Samteladze, N.; Christensen, K., DELTA: Delta encoding for less traffic for apps, Local Computer Networks (LCN), 2012 IEEE 37th Conference on, vol., no., pp.212, 215, 22-25 Oct. 2012
[5] Samteladze, Nikolai; Christensen, Ken, "DELTA++: Reducing the Size of Android Application Updates," Internet Computing, IEEE, vol.18, no.2, pp.50, 57, Mar.-Apr. 2014.
[6] Sarath Mathew, Ajaykumari K, “Improved Delta++: Smartphone Application Update Reducer”
[7] V. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals,” Soviet Physics-Doklady, vol. 10, no. 8, 1966.
[8] J. W. Hunt and M. MacIlroy, “An algorithm for differential file comparison”, scanned from Bell Laboratories Computing Science Technical Report 41 dated July 197
[9] J. Ziv and A. Lempel, “Compression of Individual Sequences via Variable-rate Coding,” IEEE Transactions on Information Theory, 24, no. 5, pp. 530-536, 1978.
[10] W. F. Tichy, “RCS - A System for Version Control,” Software: Practice and Experience, 15, no. 7, pp. 637-654, 1985.
[11] W. F. Tichy, “The String-to-String Correction Problem with Block Moves,” ACM Transactions on Computer Systems (TOCS), 2, no. 4, pp. 309-321, 1984.
[12] D. W. Jones, “Application of Splay Trees to Data Compression, “Communications of the ACM, 31, no. 8, pp. 996-1007, 1988.
[13] D. G. Korn and K.-P. Vo, “Engineering a Differencing and Compression Data Format,” Proceedings of the Usenix Annual Technical Conference, pp. 219-228, 2002
[14] D. Trendafilov, N. Memon, and T. Suel, “zdelta: An Efficient Delta Compression Tool,” Technical Report TR-CIS-2002-02, Polytechnic University, June 2002.
[15] J. Gailly, “Zlib Compression Library”. Available at http://www.gzip.org/zlib/.
[16] J. MacDonald, “File System Support for Delta Compression,” Master’s thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 2000.
[17] A. van Hoff and J. Payne, “General Diff Format Specification,” August 21, 1997. URL: http://www.w3.org/TR/NOTE-gdiff-19970901
[18] B. S. Baker, U. Manber, and R. Muth, "Compressing Differences of Executable Code," ACMSIGPLAN Workshop on Compiler Support for System Software (WCSS), pp. 1-10. 1999.
[19] C. Percival, “Naive Differences of Executable Code,” draft paper dated 2003 and available from: http://www.daemonology.net/bsdiff.
[20] J. Seward, “The bzip2 and libbzip2 Official Home Page,” 2013. URL: http://www.bzip.org.
[21] B.C. Housel, G. Samaras, and D. B. Lindquist, “WebExpress: a Client/Intercept Based System for Optimizing Web Browsing in a Wireless Environment,” Mobile Networks and Applications, 3, no. 4, pp. 419-431, 1998.
[22] J. Mogul, F. Douglis, A. Feldmann, and B. Krishnamurthy, “Potential Benefits of Delta Encoding and Data Compression for HTTP,” ACM SIGCOMM Computer Communication Review, vol. 27, no. 4, pp. 181-194, ACM, 1997.
[23] J. Mogul, B. Krishnamurthy, and F. Douglis, “RFC 3229: Delta encoding in HTTP,” IETF RFC, 2002.
[24] “Binary Delta Compression”, Microsoft, March 2004.Available at http://www.microsoft.com/en- us/download/details.aspx?id=1562
[25] “Delta Compression Application Programming Interfaces,” Microsoft, October 2009. URL: http://msdn.microsoft.com/en-us/library/bb417345.aspx



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY