International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 9 (2017) pp. 2071-2078

Efficient Big Data Transfer Technique for Static Routing Networks

Issam Alhadid[1]
University of Jordan
Business Information Systems Department
Jordan.

Sufian Khwaldeh
University of Jordan
Business Information Systems Department
Jordan.

Khalid Kaabneh
Amman Arab University,
Computer Science Department
Jordan.

Hassan Tarawneh
Amman Arab University
Computer Science Department
Jordan.

[1]ORCID: 0000-0002-5536-4333

Abstract

The era of the networks technology encourages sharing scattered information and computing capabilities, limited network bandwidth in communication and memory space needed to store and transfer big data files between client, server and vice versa, which forms a problem that cannot be solved by upgrading the networks' hardware or using current available data transfer protocols. This research suggests and proposes a new novel technique that analyzes parameters related to network technologies and big data files and utilize Huffman Encoding compression algorithm to minimize the files' size and optimize the network performance in the most effective way for static routing networks. The results of this research shows that the time needed to reduce the big data files' size using Huffman Encoding compression algorithm and the time to transfer the compressed files is less than the time calculated to transfer the original big data files in some of the network technologies. Thus, the results of the simulation shows the proposed technique in this research has minimized the transmission time for some technologies and increased the network throughput.

Keywords: Big Data, Huffman, static routing, compression, optimize, Copper, Fiber optics

INTRODUCTION

Big data is a massive volume of data and information being generated, collected or combined in different formats using new technologies such as social media, geographic information systems, Mobile data, databases and digital devices (Manyika et al., 2011; Swan, 2013; Russom, 2011). Big Data storage and transmission affected by the bandwidth and storage limitations has been raised as one of the challenges that the technologies' users are facing daily specially with emerging Mobile and Cloud technology and moving data backwards and forwards (Russom, 2011; Ammu & Irfanuddin, 2013; Tole, 2013; Nasser & Tariq, 2015). This research, proposes a technique to collect big data files and provide the appropriate decisions if it is worth to use lossless compression technique to minimize transferred files size, which ultimately will result in reducing network data traffic and reaching an optimized network bandwidth.

Data compression represents the data in minimum number of bits as possible with the aim of reducing the size of the original data, so that it either stored or transmitted (Pu, 2006; Kodituwakku & Amarasinghe, 2011; Salomon, 2004). The comparison between the different algorithms depends on the compression /decompression time and ratio factors; the compression time is needed time (in seconds) to compress a specific number of bits depending the compression algorithm.. Meanwhile, decompression time is the needed time in seconds to extract the compressed bits to its original state (reference). The compression ratio is the rate of the original uncompressed data to the rate of the compressed data (Pu, 2006; Kodituwakku & Amarasinghe, 2011; Altarawneh & Altarawneh, 2011; El-Shora et al., 2013).

The proposed technique adapted the results of Data compression techniques proposed by Song et al., (2010), Seazholtz et al., (1998), Setton& Girod (2004) and He et al. (2002). They have used a set consisting of different standard data structures. They claimed that the results of hug files compression ratio, compression and decompression time using Huffman Encoding are better than other compression algorithms. Based on that, the proposed technique adopted the Huffman Encoding algorithm in this research..

The research has been suggested to use the data compression to minimize the big data file before transferring it as messages of bits divided into packets transmitted through network physical carrier, where the network transmission time is the time needed from the first packet transmission until the last packet transmission in the message and it depends on the file size, network bandwidth, and network propagation time (Loveman et al., 2001; Hudaib et al., 2009; Song et al., 2010).

The goal of this research is to optimize the network performance and to minimize the network data traffic and big data files transmission time in the most effective way for static routing networks by analyzing the parameters related to big data and network technology based on an intelligent strategies using lossless compassion. The research uses the comparison of the results of files' transmission and propagation time using different network technologies for static routing networks to illustrate the best way to optimize the network.

RELATED WORK

Many researchers have been working to solve the problem of the big data storage, transfer, and network performance, in addition to other factors such power consumption on mobile and cloud technologies. Tierney et al., (2012) compared the performance of data movement and CPU impact using different network transfer data protocols over high latency network routs. They claimed that using Linux zero-copy system improves the TCP network performance. Yildiri et al. (2016) suggested two algorithms to enhance network performance based on the network transfer parameters; pipelining, parallelism and concurrency. The first algorithm called Adaptive PCP Algorithm sorts the files according to the bandwidth delay to maximum network bandwidth, and the second algorithm called Multi-Chunk (MC) Algorithm suggested to be used to transfer mixed of data files sizes (small and big) by dividing the dataset into chunks (Small, Mid, Large and Huge). Other researcher suggested data compression before transmission over wireless networks using special compression algorithms to reduce the power consuming on wireless networks. Kimura & Latifi (2005) claimed that power consumed by the process of compression and decompression data is much less than the power consumed by transmitting data over wireless network. Also, Park et al., (1996) stated that the relationship between the file size and network traffic will not be affected in case of changing the network topology, resources, distribution of interval time of traffic maxing.

Accordingly, the file size need to be changed or network bandwidth and throughput should be expanded. The proposed technique uses the results of data compression techniques proposed by Song et al., (2010), Seazholtz et al., (1998), Setton,& Girod (2004) and He et al. (2002). They have used a set composed by different standard test and different data structures. They claimed that results of compression ratio using Lempel–Ziv–Welch (LWZ) showed a great saving percentage, but the drawback of the compression time for large files makes it hard to use LWZ algorithm to solve big data files. On the other hand, the compression and decompression time using Shanon Fanc and Huffman Encoding are almost the same and are with best results for all files' sizes, but the compression ratio for small and large data files are relatively similar using Huffman Encoding and thus it is better than other algorithm which makes it the best choice to be used with our technique. The research will use the average of the files' size, compressed files' size, compression time and decompression time of the Huffman Encoding published in Song et al., (2010), Seazholtz et al., (1998), Setton,& Girod (2004) and He et al. (2002) Researches, the average of the result are shown in the Table (1).

Table 1: Average of the Huffman Encoding Compression Result (Kodituwakku & Amarasinghe, 2011; Somefun & Adebayo, 2004)

Avg. of original Files' Size	Avg. of com. file's size	Avg. of Compression Ratio	Avg. of Compression Time	Avg. of decompression Time
2,466,475 Byte	813,739 Byte	67%	133.333 ms	133.333 ms

Table (2) shows the average rates of data transmission speed using different network technologies according to Song et al., (2010), Seazholtz et al., (1998), Setton,& Girod (2004) and He et al. (2002). The rates will be used to compare the original and compressed Big Data files' transmission time using different network technologies and transmission rates (Song et al., 2010; Seazholtz et al., 1998; Setton, & Girod, 2004; He et al., 2002).

Table 2: Network technologies' bandwidth and throughout (Song et al., 2010; Zhanga et al., 2015; Crayford, 1995; Ata et al., 2001; Papagiannaki et al., 2003)

Network Technology	Network Bandwidth	Network Throughput
ADSL Lite	1.5 Mbits/s	1.5 Mbits/s
E1	2.048 Mbits/s	1.984 Mbits/s
T3/DS3	44.736 Mbits/s	44.21 Mbits/s
Fast Ethernet	100 Mbits/s	100 Mbits/s
OC3	155 100 Mbits/s	150 Mbits/s
OC12	622 100 Mbits/s	599 Mbits/s
OC48	2.5 Gbits/s	2,396 Mbits/s
OC192	9.6 Gbits/s	9,585 Mbits/s
10 Gigabit Ethernet	10 Gbits/s	9,585 Mbits/s

Based on compression algorithms, researches published by Song et al., (2010), Seazholtz et al., (1998), Setton,& Girod (2004) and He et al. (2002) claimed that to compress 2,466,475 byte KB, it will need 266.666 millisecond for compression and decompression time to reduce the file size to 813,739 byte. Accordingly, the time needed to compression / decompression and the compressed files' transmission time will be calculated, which will be compared with the original file transmission time using the transmission speed for data through different network technologies. Our technique will be implemented on static network route with 20 nodes scattered, where the network routes are directly connected and manually configured. The maximum number of hops for each transition will not exceed 16 hops (Jin et al., 2003) while the distance between the two nodes is between 0.5 mile and 500 mile (Hura & Singhal, 2001).

PROPOSED TECHNIQUE

This research considers the problem of data transfer big data files as a challenge that can't be solved by upgrading or replacing the networks' hardware. This research proposes a technique based on Huffman Encoding compression to reduce the size of the big data files size. The technique uses the compassion between original and compressed big data files transmission time affected by propagation factors and networks technologies used by most of internet consumers and ISPs. Figure (1) shows the proposed technique steps.

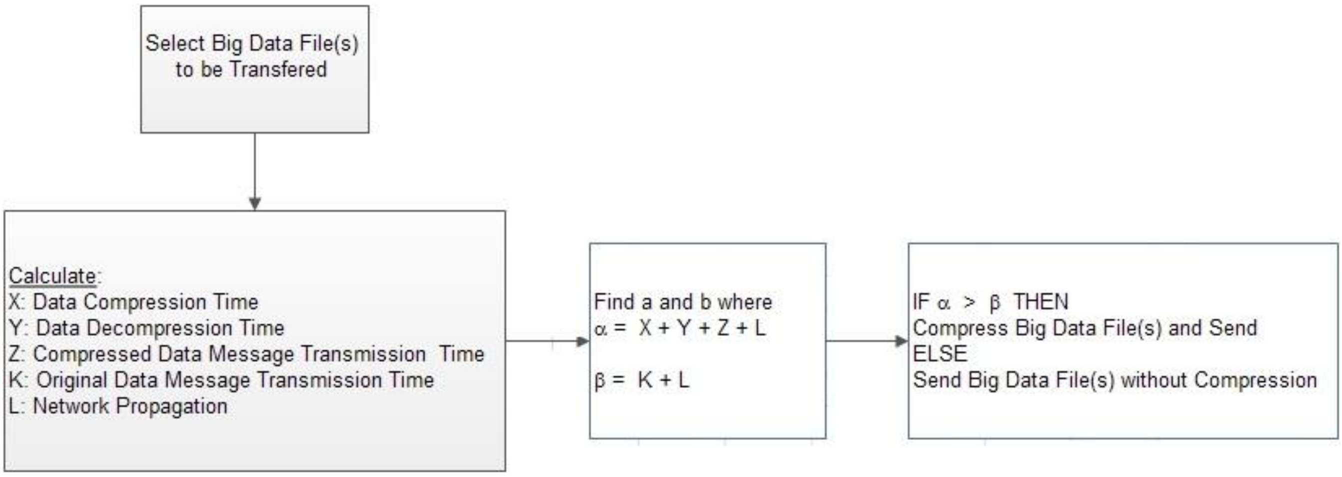

Figure 1: Proposed Technique

Proposed technique steps:

1. Select big data files to be transferred: Big Data File(s) might be selected by the users to be transferred when a network is available, or when it has been processed, collected, sorted, classified or encrypted by a system.
2. Calculate the compressed size and time: in this step, the technique will calculate the size of selected big data files, including the compression and decompression time.
3. Check the transmission available technology: check the available network technology and bandwidth. Calculate the total transmission time for the original and the big compressed data files based on the network technology bandwidth and network propagation using the following parameters**:** files' size, network throughput, link propagation delay, packet transmission delay, number of hops, queuing delay, number of packets for each message and per hop processing delay (Jain et al., 2004; Papagiannaki et al., 2003). To calculate the transmission propagation time, the transmission time parameters listed in Table (3) will be used in addition to the following equations:

Network transmission time = data transmission time + transmission propagation time

Propagation delay + transmission delay + queuing delay + processing delay

Where:

Propagation delay = number of hops * link propagation delay

Link propagation delay = length of physical link / propagation speed in medium

Transmission delay = message size (number of packets) * packet transmission time

Queuing delay = (number of hops - 1) * packet transmission delay

Processing time = (number of hops - 1) * per-hop processing delay

Table 3: Transmission time parameters

parameter	**Value**
Per-hop processing delay	0.0000025s (2.5μs)
Propagation speed in medium (copper)	62094.132 mi/s
Propagation speed in medium (Fiber optics)	186282.397 mi/s
Message size	64 kb
Data file size	Byte
Length of physical link	Mile
Number of Hops	1-16

1) Check if big data files compression process is feasible to be proceeded to the next step according to the big data files size, compression time, decompression time, data transmission time and network propagation time. All factors mentioned should be less than the original big data files network transmission time and network propagation time without compression. If the process of compression will reduce network transmission time, then compress files and send. Otherwise, just send the original

big data files without compression. As described in Figure (1), the research compared compressed big data files related transmission information (α) with (β) which is the information related to the original big data files transmission time, consequently, if α is less than β, then it is feasible to proceed in comparison process and send the data, otherwise, you don't need to compress the big data files and just send it without any future processing.

If the selected data might pass the evaluation of α and β, then it is feasible to be compressed before sending because it will reduce the network traffic and help solving the problem of limited network bandwidth and the transfer of big data files.

The following assumptions shown in table (4) have been adopted in the simulation of the proposed technique to generate a static routing network topology where source and destination nodes for each transaction are randomly chosen, and then the system will find the shortest distance and minimum number of hops between source and destination according to the stored information related to generated network topology.

Table 4: Configuration of the Transmission Simulation

Parameter	Value
Number of nodes	20
Number of Runs	11
Number of Transactions/Run	1,000
Total number of Transactions	11,000
Minimum Number of hop	1
Maximum Number of hop	16
Hop distance lower bound	0.5 mi
Hop distance upper bound	500 mi
Minimum Big Data File size	1 TB
Maximum Big Data File Size	30 TB
Compression rate	67%.

Table (4) shows the simulation's big data files' sizes according to the research assumptions. The simulation has run 11 times, where each run executed 1,000 transactions with different number of hops, diverse big data files' sizes and various networks' technologies. The table demonstrates the size of the original and compressed big data files where the original big data files' size of all transactions is 54740 Terabyte. On the other hand, the size of the compressed big data files used in Huffman Encoding compression is 18064.2 Terabyte with 67% compression ratio.

SIMULATION RESULTS

The simulation results of the proposed technique provide information about how different network technologies affect the performance of the network throughput and transmission time as well as the network prorogation time, which are in turn affected by the size of the data file, physical distance, speed in medium and number of hops and other factors.

The comparison between original and compressed data transactions' sizes classified according to simulation runs is shown in Figure (2). Even though that the size of the compressed big data files is less than the size of the original big data files based on the 67% Huffman Encoding compassion ratio, as illustrated in Figure (2), the proposed technique will show that the time required to transfer the data not only depends on the file size, but also it depends on other important network related factors, such as network bandwidth and propagation time parameters.

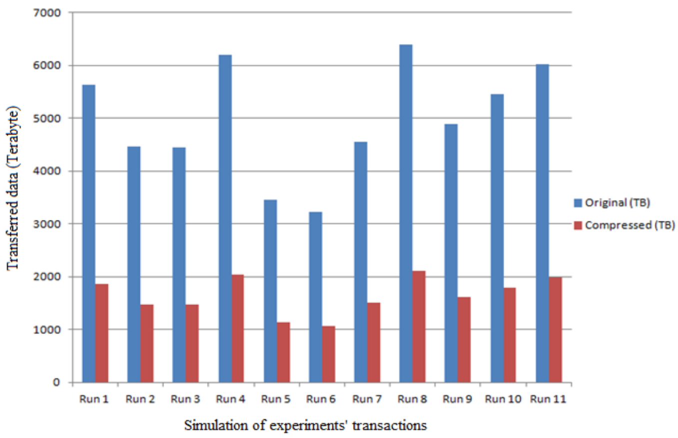

Figure 2: Comparison between original and compressed data transactions' sizes classified according to simulation runs

Figure (3) shows the comparison between the original and compressed transmission time without the network propagation time. Thus, the results shown are based on the network throughput. Moreover, the Figure shows a great transmission time saving when applying the proposed technique with ADSL Lite, E1 and E3 technologies.

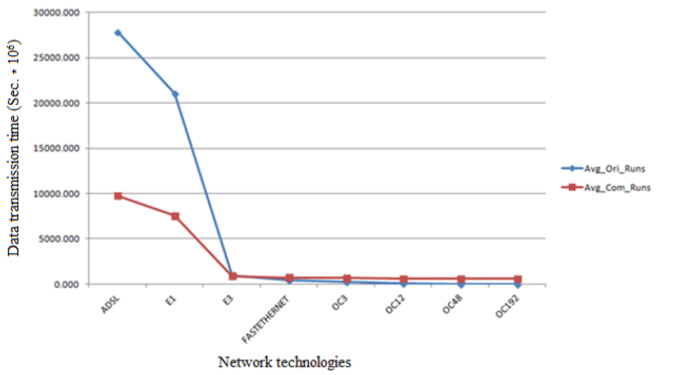

Figure 3: Comparison between the average of the 11 runs transmission time for original and compressed big data files without network propagation time.

Transmission time used in the proposed technique with ADSL Lite technology will save more than 50% of the transmission time if compared to the original Big Data File(s) without compression. On the other hand, the results of transmission original data using Fast Ethernet, OC3, OC12, OC48 and

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 9 (2017) pp. 2071-2078

OC192 technologies are less than the result of transmission time for the compressed data affected by the long time needed to compress and decompress the big data files.

Network propagation time depends on different network related parameters. The most important parameters that significantly and directly affect the propagation time are the files size, network technology, network bandwidth and type of the carrier medium (Copper or Fiber optics). The files' size, network technology, bandwidth and the distance among the related parties are directly proportional to network propagation time, in addition to the type of the carrier medium where the propagation speed in medium of the Copper is one-third of the Fiber optics medium. Therefore, it means that the speed of the Fiber optics is faster three times more than the Copper medium.

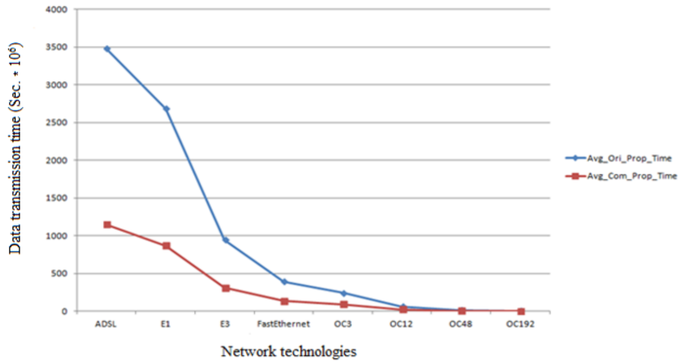

Figure 4: Comparison between original and compressed big data file network propagation time

The results shown in Figure (4) illustrate the average of 11 runs' network propagation time for packets hops classified based on the network technology. The per-hop processing delay is 0.0000025s (2.5µs) (Papagiannaki et al., 2003), which is not a significant parameter. The maximum value of processing time for the maximum number of hops (16 hops) is 0.00005s (15µs). Figure (4) shows a high result of network propagation time using ADSL Lite technology as a result of the carrier medium speed (Copper) and the slight network bandwidth. Even though E1, E2 and Fastethernet are using the same medium used with ADSL Lite technology, the propagation time is less than ADSL Lite as a result of the better bandwidth of the mentioned technologies. High speed technologies based on Fiber optics with high bandwidth and carrier medium speed decrease to the network propagation time. The higher the bandwidth and/or carrier medium speed, the less network propagation time.

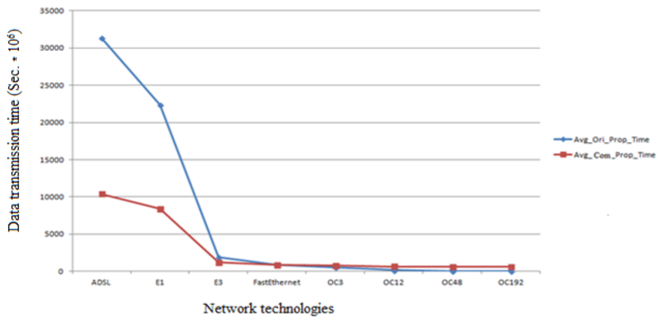

Figure 5: Comparison between original and compressed big data file network transmission time

Figure (5) shows the comparison between the original and compressed total network transmission time including data transmission time and network propagation time. The results by using the proposed technique with Copper based technologies are effective and show an improvement to the network throughput, such technologies are ADSL Lite, E1, E3. Thus, this research recommends the Copper based technologies to be used with our proposed technique after the network propagation time has been added which proves that the total transmission time using Huffman Encoding compression is less than the network transmission time needed by original big data files. On the other hand, this research does not suggest the implementation of the proposed technique with Fastethernet, OC3, OC12, OC48 and OC192 which are technologies based on the Fiber optics because it is not feasible. The total network transmission time needed to transmit the original big data file is less than the time required to transmit the same compressed data.

SIMULATION RESULTS & STATISTICAL ANALYSIS

Figure (6) presents the statistical analysis steps in this research, which start from gathering the results until drawing the conclusion. However, after doing the experimental results, there is a need to define the following terms: need to be defined:

- Min: is the minimum transmission time with propagation.
- Max (worst): is the worst Transmission time with propagation.
- Mean (average): is the average Trans time with propagation.
- Std: is the standard deviation.

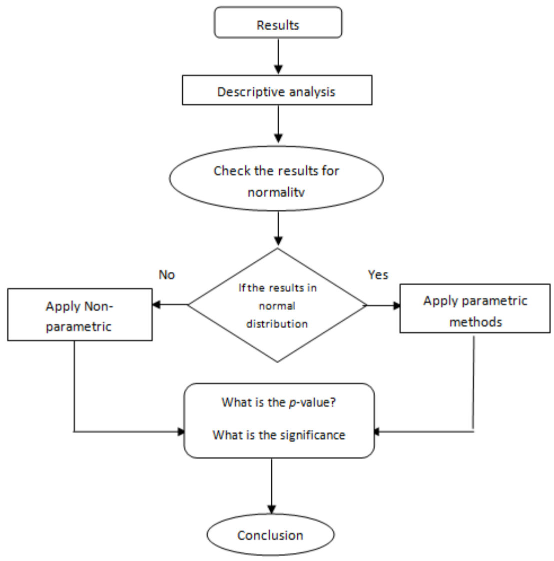

Figure 6: Research Results Statistical Analysis Steps

Firstly, for the normality distributions, results has been tested in terms of checking the underline parameters distribution, where the parametric procedure (normal) care about the underline results, whilst the non-parametric procedure will be ignored. To check the normality distribution, the research used the Shapiro-wilk test with 0.05 critical levels to test the distribution of the results (see table 5). To make a fair test, 16 results has been selected randomly that were obtained by the proposed technique for both original and compressed big data files.

Table (5) shows that the Kolmogorov-Smirnov and Shapiro-Wilk significant level (sig.) is greater than the critical level 0.05, then the results are normal distributed. However, regarding the normality test in this research, pared sample test (T-test) has been used. T-test is parametric version of a paired sample t-test, where the value of alpha is 0.05. The significance testing helps to make a judgment about a claim by addressing the question: Can the observed difference be attributed to chance?

Table 5: Tests of Normality (Kolmogorov-Smirnov and Shapiro-wilk tests)

Data	Kolmogorov-Smirnov[a] Sig.	Shapiro-Wilk Sig.
Org_ADSL	.200*	.531
Org_E1	.200*	.626
Org_E3	.200*	.531
Org_ETHERNET	.200*	.531
Org_OC3	.200*	.531
Org_OC12	.200*	.531
Org_OC48	.200*	.532
Org_OC192	.200*	.527
Com_ADSL	.051	.045
Com_E1	.200*	.531
Com_E3	.200*	.531
Com_ETHERNET	.200*	.531
Com_OC3	.200*	.531
Com_OC12	.200*	.531
Com_OC48	.200*	.561
Com_OC192	.200*	.531

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

In order to conduct the test, there are two important numbers (Values). One number is the p-value of the tests statistics, which represents the probability that the observed statistic occurred alone by chance. Whilst the other number is the significance level (alpha), both numbers are between zero and one. Alpha represents a value that p-values measures against and it gives us the probability of a type 1 error. Moreover, it associates with the confidence level test (e.g. if the confidence level for the results is 90%, then the alpha value is 1− 0.90 = 0.10). In general, for the results with C% confidence level, alpha value is $\alpha = 1 - C/100$.

The test statistic p-value for sample data is a way of showing how extreme that statistic is. Namely, the smaller p-value is the observed outcome is more statistically significant. In order to determine the statistical is significance of the observed outcome sample data, p-value has been compared with alpha as follows:-

- If p-value is less than or equal alpha, then the results are statistically significant.
- If p-value is greater than alpha, then the results are not statistically significant.

The p-value statistical results obtained by eight techniques on two kinds of data (original and compressed) are zero. The results confirm that there are statistical significant differences between original and compressed big data files on (ADSL Lite, E1 and E3) ($P\text{-value} < 0.05$). Compressed data proved to be better than original by obtaining less network transmission time. Otherwise, original data proved to be better than compressed one on Ethernet, OC3, OC12, OC48 and OC192.

CONCLUSION

Results of the research simulation show that the proposed technique will minimize the big data files transmission time for some of the network technology adopted. Results of the simulation using the proposed technique for all network technologies applied in our research demonstrate that the time needed to transfer, compress and decompress big data files using Huffman Encoding compression algorithm are less than the time calculated to transfer the original big data files using the technologies based on Copper such as ADSL Lite, E1, and E3. Meanwhile, if the network is based on the other high speed technologies such as Fastethernet, OC3, OC12, OC48 and OC192 which are based on Fiber optics, the research suggests to send data without compression and decompression processes because of the high compression/decompression processing time which might be replaced using super computers or using other improved compression algorithm. Accordingly, using the proposed technique with Copper based technologies will increase the networks throughput, optimize the network performance and minimize the network data traffic and transmission time in the most effective way for static routing networks.

FUTURE WORK

The results of transmission original data using Fastethernet, OC3, OC12, OC48 and OC192 technologies is less than the result of transmission time of the compressed data as a result of the processing time required to compress and decompress the big data files. Thus, in case of improving or replacing the Huffman Encoding with a faster compression technique, the result might be promising. Also, implement the proposed technique with distributed query processing. In addition to that, apply the proposed technique with dynamic and mobile networks and compare the result according to the transmission time, mobile power consumption and other parameters.

REFERENCES

[1] Altarawneh, H., & Altarawneh, M. (2011). Data compression techniques on text files: A comparison study. *International Journal of Computer Applications*, *26*(5), 0975-8887.

[2] Ammu, N., & Irfanuddin, M. (2013). Big Data Challenges. *International Journal of Advanced Trends in Computer Science and Engineering*, *2*(1), 613-615.

[3] Ata, S., Takine, T., Murata, M., & Miyahara, H. (2001). Performance improvement of ABT protocols with combined bandwidth/buffer reservation. *Performance Evaluation*, *46*(4), 255-274.

[4] Crayford, I. (1995). "Fast Ethernet" gets plug-and-play. In *WESCON/'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies'* (p. 354). IEEE.

[5] El-Shora, D. A. S., Mohamed, E. R., Lashin, N. A., & El-Henawy, I. M. (2013). PERFORMANCE EVALUATION OF DATA COMPRESSION TECHNIQUES VERSUS DIFFERENT TYPES OF DATA. *International Journal of Computer Science and Information Security*, *11*(12), 73.

[6] He, E., Leigh, J., Yu, O., & DeFanti, T. A. (2002). Reliable blast UDP: Predictable high performance bulk data transfer. In Cluster Computing, 2002. Proceedings. 2002 IEEE International Conference on (pp. 317-324). IEEE.

[7] Hudaib, A., Kaabneh, K., & Qatawneh, M. (2009). Modified on-demand multicasting routing protocol for ad hoc networks. *WSEAS Transactions on Information Science and Applications*, *6*(2), 147-158.

[8] Hura, G. S., & Singhal, M. (2001). *Data and computer communications: networking and internetworking*. CRC Press. pp. 18, pp.

[9] Jain, S., Fall, K., & Patra, R. (2004). *Routing in a delay tolerant network* (Vol. 34, No. 4, pp. 145-158). ACM.

[10] Jin, C., Wang, H., & Shin, K. G. (2003). Hop-count filtering: an effective defense against spoofed DDoS traffic. In Proceedings of the 10th ACM conference on Computer and communications security (pp. 30-41). ACM.

[11] Kimura, N., & Latifi, S. (2005). A survey on data compression in wireless sensor networks. In *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II* (Vol. 2, pp. 8-13). IEEE.

[12] Kodituwakku, S. R., & Amarasinghe, U. S. (2010). Comparison of lossless data compression algorithms for text data. *Indian journal of computer science and engineering*, *1*(4), 416-425.

[13] Loveman, J. S., Allen, M. S., White, R., & Haynes, C. E. (2001). *U.S. Patent No. 6,211,869*. Washington, DC: U.S. Patent and Trademark Office.

[14] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity.

[15] Nasser, T., & Tariq, R. S. (2015). Big Data Challenges. *J Comput Eng Inf Technol 4: 3. doi: http://dx. doi. org/10.4172/2324*, *9307*, 2.

[16] Papagiannaki, K., Moon, S., Fraleigh, C., Thiran, P., & Diot, C. (2003). Measurement and analysis of single-hop delay on an IP backbone network. *IEEE Journal on Selected Areas in Communications*, *21*(6), 908-921.

[17] Park, K., Kim, G., & Crovella, M. (1996, October). On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Network Protocols, 1996. Proceedings. 1996 International Conference on* (pp. 171-180). IEEE.

[18] Pu, I.M., 2006, Fundamental Data Compression, Elsevier, Britain. 3-5

[19] Russom, P. (2011). Big data analytics. *TDWI Best Practices Report, Fourth Quarter*, 1-35. (big data size)

[20] Salomon, D. (2004). *Data compression: the complete reference*. Springer Science & Business Media.

[21] Seazholtz, J. W., Sims, W. N., & Sistanizadeh, K. (1998). U.S. Patent No. 5,812,786. Washington, DC: U.S. Patent and Trademark Office.

[22] Setton, E., & Girod, B. (2004). Congestion-distortion optimized scheduling of video over a bottleneck link. In Multimedia Signal Processing, 2004 IEEE 6th Workshop on (pp. 179-182). IEEE.

[23] Somefun, M., Adebayo, O., (2014), Evaluation of Dominant Text Data Compression Techniques, *International Journal of Application or Innovation in Engineering & Management (IJAIEM),* 3(6), 162-169

[24] Song, H., Kim, B. W., & Mukherjee, B. (2010). Long-reach optical access networks: A survey of research challenges, demonstrations, and bandwidth assignment mechanisms. *IEEE communications surveys & tutorials*, *12*(1).

[25] Swan, M. (2013). The quantified self: Fundamental disruption in big data science and biological discovery. *Big Data*, *1*(2), 85-99.

[26] Tole, A. A. (2013). Big data challenges. *Database Syst J*, *4*(3), 31-40.

[27] Tierney, B., Kissel, E., Swany, M., & Pouyoul, E. (2012, October). Efficient data transfer protocols for big data. In *E-Science (e-Science), 2012 IEEE 8th International Conference on* (pp. 1-9). IEEE.

[28] Yildirim, E., Arslan, E., Kim, J., & Kosar, T. (2016). Application-level optimization of big data transfers through pipelining, parallelism and concurrency. *IEEE Transactions on Cloud Computing*, *4*(1), 63-75.

[29] Zhanga, S., Takeshita, H., Okamoto, S., Yamanaka, N., Zhanga, S., Takeshita, H., & Yamanaka, N. (2015). Energy Efficient and Enhanced-type Data-centric Network Architecture. *International Journal of Computer & Information Science (IJCIS)*, *16*(1), 60-70.