Journal of Engg. Research Vol. 6 No. (3) September 2018 pp 54-69

SNT algorithm and DCS protocols coalesced a contemporary hasty file sharing with network coding influence

Suthir S* and Janakiraman S**

**Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India and working as Assistant Professor, Dept. of CSE, Panimalar Engineering College, Chennai, Tamilnadu, India.*

***Assistant Professor, Banking Technology (An Inter-disciplinary of Computer Science & Engineering and Management), Pondicherry University, jana3376@yahoo.co.in*

**Corresponding Author: suthirsriram@gmail.com*

ABSTRACT

Network coding has presently drawn marvelous attention and gifted enrichment of routing to improve network throughput. In this paper, we propose a swift network transfer (SNT) algorithm and deduplication, caching, and security (DCS) protocols to enhance the performance of peer-to-peer fast file sharing with network coding that can attain 10% to 15% privileged throughput when compared to that of other P2P multicast networks. Initially, it is proposed to find peer nearness on the network using swift network transfer (SNT) algorithm. Secondly, the functions of protocols used in the proposed concept are analyzed, where, in the first protocol, there is deduplication to abolish replica photocopy of files for maintaining the database storage space effectively. The second protocol network caching enables us to find file request node in data flow. The last protocol is command and control performances to provide security for checking IP address and mismatch of uplink and downlink transmission nodes.

Keywords: Fast file sharing, SNT, P2P, DCS, network file sharing.

INTRODUCTION AND RELATED WORK

P2P is a gifted innovation when compared to that of actualized multicast at the application layer, where beneficiaries (peers) receive information/file and promote information as well. In general, network communications are traded by routing over moderate nodes among the starting place to end place. The idea of network coding (NC) was initially presented by Ahlswede et al. (2000).

The traditional network file sharing is used for file transfer from source to destination as it is just forwarded information to the next node. It will not change the way or modify and force any coding part. Using NC, file sharing is exchanging documents that it allowed to apply coding on information or data. Also, it will gather outward messages that may not be similar to inward messages at any given transitional node. Ahlswede et al. (2000) initially presented the NC and demonstrated the handiness of NC. Lun et al. (2006) demonstrated the effectiveness of NC on remote network system frameworks, so that examining NC was centered on wire connected networks (Ho et al., 2006; Katti et al., 2008; Lun et al., 2006). Likewise, utilizing network coding as a part of peer-to-peer was primarily proposed in Gkantsidis & Rodriguez (2005), and latest feasibility studies on NC in genuine test beds were executed in Wang & Li (2007).

P2P networks are a flawless place to apply NC because of two motives: Initially, the topology of a distributed/P2P framework has developed self-assertively; it is anything but difficult to mould the topology to support NC. Secondly, the nodes in a shared P2P framework can do overwhelming processes, for example, encoding and decoding, not just receiving and sending files/data (Yang & Yang, 2014; Wang & Ansari, 2011). NC refers to a plan wherever a node is permitted to create output data/files by encoding (i.e., registering certain elements), it receives data/files. In this way, organizing coding permits data to mix, as opposed to the traditional directing methodology where every node just sends the received messages (Suthir & Janakiraman, 2014).

Structured network is circulated peer-to-peer; it gives the dispersed ordering to mapping substance and associates peers in a solitary address space (e.g., Kademlia and eDonkey file sharing applications). The principle downside requires complex routing algorithm and high cost of network safeguarding to handle support and withdrawal of companion peers (Zhao et al., 2014). The unstructured network gives a benefit message and keeps up P2P by wilful support of peers without the main server (e.g., Bit Torrent, Gnutella, and Kazaa). The disadvantage is that it creates high traffic in the network due to proficient utilization and pursuit of disseminated network resource.

The researchers have configured the network with centralized peer-to-peer server to manage all peers and share the files that provide continuous service to support the efficient file searching and storing of each peer. The disadvantage of not centralized peer-to-peer was incrementing the server stack that causes the extensibility problem, and a problem of the main server can influence the whole network (Chou, Wu & Jain, 2004; Darlagiannis, Heckmann & Steinmetz, 2006).

The state of affairs of content distribution comprising a solitary inception server associated with a mutual bottleneck connection to various clients each is furnished with limited memory of cache (Suthir & Janakiraman, 2013; Suthir & Janakiraman, 2016). The clients' concern is a succession of substance solicitations from a position of prevalent data, then objective of cache work, and also the server to such an extent that these solicitations are happy with the amount of bits driven over the mutual connection. So we decided to utilize the bottleneck obsession to stay away from the overburden, and focal database is kept up with SNT algorithm and DCS protocols for deduplication, encoded caching, and providing high security on transactions time. At that point, why do not we decide to proceed with centralized peer-to-peer network? As per the proposed setup of the brought together centralized peer-to-peer networks, they do not depend on any file sharing application like Gnutella, eDonkey, and so forth. In this paper, we decided to propose a SNT algorithm and DCS protocols to implement, maintain, and get higher throughput effectively.

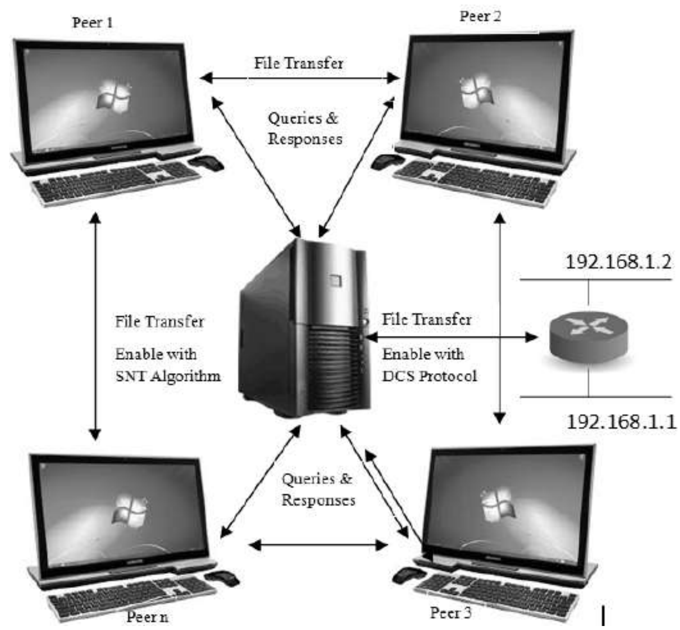

Fig. 1. Centralized peer-to-peer network file sharing system.

The following is the structure of this article: The paper starts with an explanation of the problem definition. Following that, the techniques used in this paper are presented and followed by sub-sections. Next, SNT algorithm and its concepts are described, which is pursued by deduplication and file searching protocol, with its contributions to the proposed concepts being explained. Caching with security as encoding as well as decoding processes are described in the next section. Security with command and control protocol performances for offering security to file transactions is explained in the following section. In the next section, various parameters used for network file sharing system are represented in tabular format. The paper ends with a conclusion and future work section.

PROBLEM DEFINITION

There are various problems related to file sharing system including scalability and performance, reliability, and anonymity. Problems linking to the performance concerns come up from various servants relating to the network through low speed modems. These servants are generally scattered throughout the network, and they are over flooded with messages until they turn into unresponsive that is no more varied from them being offline. This makes the network greatly fragmented, hence causing query messages to be restricted and not to exceed each grouped fragment. This corrupts the search outcomes drastically. Also, the download failures have been caused through the restriction of upload speed or partial files' sharing through the users who share those files (Ding, Nutanong & Buyya, 2005).

To share the files in P2P network, fast and securely with the influence of network coding in peer-to-peer file sharing networks, the research scheme is proposed as follows:

i) Swift Network Transfer (SNT) Algorithm–for effective peer joining.

ii) Deduplication–for fast file searching.

iii) Caching–to improve caching.

iv) Security–to share files with Command and Control.

By implementing the above-mentioned algorithm and protocols, this technique can attain good higher throughput than other peer-to-peer network file sharing systems. However, there are several studies concentrating on the peer-to-peer file sharing system. There are no other studies that focus purely on estimating fast file sharing in peer-to-peer system. Hence, this study acts as a unique analysis that aims to focus on investigating a contemporary fast file sharing system in a network.

Table 1. Parameter Definitions

S.No	Symbol	Description
1	SNT	Swift Network Transfer
2	DCS	Deduplication, Caching with Security
3	P2P	Peer to Peer
4	APT	Advanced Persistent Threat
5	DNS	Domain Name Service
6	PID	Peer File Identifier
7	S, D	Space_filling curve, Hilbert_curve
8	f, i, d	File Key, Node, Distance
9	MBD	Malicious DNS Detector
10	SBD	Signature Based Detector
11	ABD	Anomaly Based Detector
12	C&C	Command and Control
13	DOS	Denial of Service
14	LAN	Local Area Network
15	DHT	Distributed Hash Table

RELATED LITERATURE

There are some state-of-the-art techniques, which are considered to be most significant in previous analyses. Javanmardi, Shojafar, ShariatmadarI & Ahrabi (2014) have proposed a new technique called PGSW OS (P2P grid semantic web OS) technique. This developed technique is mainly based on semantic technology and P2P grid architecture in order to enhance resource management in internet operating systems by resource discovery using semantic features. This technique has integrated semantic overlay networks and DHTs (distributed hash tables) in order to enable semantic based resource management through advertising resources in DHT depending on their annotations to allow semantic based resource matching. The findings have suggested that the proposed technique has decreased the computational complexity in web OS atmosphere. Also, it provides better search expressiveness, precision, scalability, and improved resource usage.

Javanmardi, Shojafar, ShariatmadarI & Ahrabi (2015) have proposed a novel technique, FR TRUST (fuzzy reputation trust based technique), for trust management in the semantic P2P grids. A peer-to-peer is the special type of grid systems, where P2P communications are employed for interaction among trust management and nodes. The authors have employed fuzzy theory in trust overlay network termed as FR TRUST, which designs the structure of network and storage of reputation information. Also, they have presented computation system and reputation collection for the semantic P2P grids. This system has used fuzzy theory to estimate peer trust levels that may be high, medium, or low. The findings have shown that FR TRUST technique merges better computational capability and high ranking accuracy.

PROPOSED ALGORITHM

Swift Network Transfer (SNT) Algorithm Node Vicinity Illustration

A landmark strategy is easily utilized for characterizing node nearness in the network (Shen & Xu, 2008; Ratnasamy et al., 2002). It depends on instinct whether nodes that are closer to each other will probably have comparable separations to the couple chosen landmark nodes. It is expected to have m landmark nodes that are randomly spread over the network. Every node calculates the substantial distance between m landmarks and the vector of space <d1; d2; . . . ; dm> since it is facilitated in the Cartesian space. There are 2 lively near nodes resulting in comparative vectors. Space filling curves (Asano et al., 1997) and the Hilbert curve (Xu, Mahalingam. & Karlson, 2003) are utilized in order to delineate the m dimensional landmark vectors into genuine statistics, such that the nearness connection between the nodes is safeguarded. The genuine statistics is considered to be the Hilbert statistics of the node indicated with D and space-filling curve as S.

Node attention illustration: Stable hash functions, for example, stable hash functions-1, are broadly utilized as part of the distributed hash table networks for peer file ID (PID) because of their safe condition and impact. At the point when utilizing similar hash function, it becomes practically impossible to discover 2 unique messages that create a similar message process. Stable hash function compels to group messages in view of message distinction.

Grouping substantially near and general attention nodes: To recognize nodes in various live locations and utilizations cyclic lists and to additionally group lively near nodes in light of their concerns.

Structure and preservation: When node i joins the framework, then nodes by PIDs equivalent to (Si%d; Di) are present, and when node i is considered as a consistent node, then it becomes a user in sub-cluster (Leu, Yu & Yueh, 2015; Shen & Liu, 2015). On the off-chance, when node i is considered as a super node, then it becomes reinforcement in support of the server node in the sub-cluster. Before server is abandoned, unique reinforcement swaps the parting server. In the action when there is no node with PIDs equivalent to (Si%d; Di) and node i will be considered as a super node, then it becomes a server node of the sub-cluster, and recently the connected nodes by PIDs as (Si%d; Di) will interface with the node. On the off-chance, when node i will not be considered as a super node, then it briefly performs the works of a server in the anticipation of joining a super node that is obtainable to replace the server.

Algorithm 1 SNT algorithm for Search, Join, and Replace in Network Peers.

Creating Network Peers File ID:

Step 1:- $PID_1 = (S_1, D_n)$, $PID_2 = (S_2, D_n)$, …

$PID_m = (S_m, D_n)$

Search the Nearest PID with Server:

Step 2:- for (i=0; i<m; i++) do

Step 3:- S_i = n.call_on_Server (PID_i);

Step 4:- if n is a normal Peer Node then

Step 5:- Consider S_i as Server

Joining the Peers:

Step 6:- Server $PID_i = S_i$;

Step 7:- Server PID_i . link (n,l);

Step 8:- else if S_i is a super node then

Step 9:- S_i include_to_backup_list(n);

Replacing the Peers:

Step 10:- else if S_i in the sub_cluster of interest i then

Step 11:- Get super node from its backup list to replace alone

Step 12:- Alert clients concerning the server change

Step 13:- else alert clients to rejoin in to the structure

Step 14:- end if, end if, end if, end for

In the event that one of the nearer nodes neglects headed for react amid a specific time period T, the server discovers and interfaces another novel nearer node. During a sub-cluster, the server chooses an auxiliary server commencing reinforcements, which is, replacing it leading its disappointment. It additionally informs entire users regarding the optional server. Prior to the abandonment of the server, it asks for the optional server to survive the novel server and tells the entire users. Next, users associate or join the novel server to grip impact of the server disappointment on their users. In particular, every user tests its server occasionally. In the event that a user/client c does not get an answer from its servers amid T, c accepts that s be unsuccessful and joins the optional server. It additionally enhances its unwavering quality; various auxiliary servers can be utilized.

DEDUPLICATION

The prominence topic of the proposed system framework is diminution of acquired overhead progressively real time situations and environments. The procedures in the proposed system framework are file sharing fundamentally and can be divided and encrypted into fragments by utilizing deduplication method for expelling the numerous duplicates of file. The information recovery should be possible by utilizing the mystery offers from the minimized number of servers (Shojafar et al, 2015). This procedure prompts to information uprightness and label consistency and this backing for adaptation to non-critical failure and information stockpiling proficiency and, in addition, security.

Algorithm 2 Call_On File 'f' in Network Peers Algorithm

Input as File Key:

Step 1:- n.callon (key) {

Step 2:- Read file 'f' key;

Read File:

Step 3:- Read file 'f' PID (D%d; D/d);

Step 4:- if key interests then

Analyze the Duplication of File:

Step 5:- Throw application to server of sub cluster D%d;

Step 6:- if Found more number of same files removes infrequent peers file

Step 6:- if collect optimistic reply from server then exit

Step 7:- Callon (D%d; D_n);

Fast Searching File:

Step 8:- if collect weak reply then

Step 9:- Use SNT algorithm then

Step 10:- Callon (D%d; D=d);

Step 11:- end if, end if, end if

The Call_On File "f" in Network Peers Algorithm has a few phases:

i) Intra_cluster seeking (comprising of intra_sub_cluster seeking and inter_sub_cluster seeking) and

ii) Inter_cluster seeking (Distributed Hash Table routing)

On the off-chance, the intra-sub-cluster seeking is unsuccessful. The cluster depends on inter-sub-cluster seeking. In the event that the inter-sub-cluster seeking is unsuccessful, it will rely on DHT routing for file seeking. At the point when i node needs to recover a data/file, the files ID/key is one among campaigner advantage characteristics and utilizes the intra-sub-cluster seeking. The node i was sending demand in the sub-cluster server for attention. Reviewing the server keeps up the directory of entire files in sub-cluster. The server gets a demand all time; it is verified in the event that the sub-cluster is asked for data/file. On the off-chance that is located, the server throws the spot of file to the campaigner specifically. In the event that the file key is not one among the campaigner attention characteristics, i node verifies the presence of file in its group (inter-sub-cluster seeking). On the off-chance, having a copy of data/file, it ought to be put away in a sub-cluster nearest to PID (D%d; Di). Consequently, the campaigner sends a demand with (D%d; Di) as an objective. The demand is sent with servers in every sub-cluster in a campaigner group.

In the event that there is no demand for data/file, the data/file demand for routing is executed in view of cycloid routing algorithm (inter-cluster searching) (Li et al, 2015; Zhao, 2015). At that point, i node computes the PID of the file (D%d; D=d) and conveys a message of Call On (peer file identifier PID). This routing algorithm does not prompt all the more overhead. Routing between lively near nodes enormously enhances file location effectiveness.

CACHING

In network operations, there are two comprehension parts of caching on the execution of stages. Firstly, it is giving content nearby, where, in the traditional caching strategy, recreating is utilized making the demand for content near the confined. In the event that a user discovers that it is nearby, the content is served as neighbouring; the main server just sends the demand file by effortless orthogonal unit cast transaction (Maddah-Ali, 2015). On the off-chance of more than one user, then it utilizes the remote network stream, as creation of traditional cache memory is notable by putting away aggregate content in local cache memory. Also, making concurrent multicasting, it is taken afterwards to fulfil the demand of numerous users with a wide range of needs in a single multicast stream. Interpreting is empowered in a neighbourhood cache to discover demand territories in data streams, so by their original user, requests are known and planned intentionally. It is utilized to decrease the congestion of the network; it impacts the normal universal cache size in sufficiently substantial to store total content. The main server arranges the subset cache precisely, not overlap. Coordination in position stage is unrealistic to dispose rate reduction (Pedarsani & Maddah-Ali, 2016).

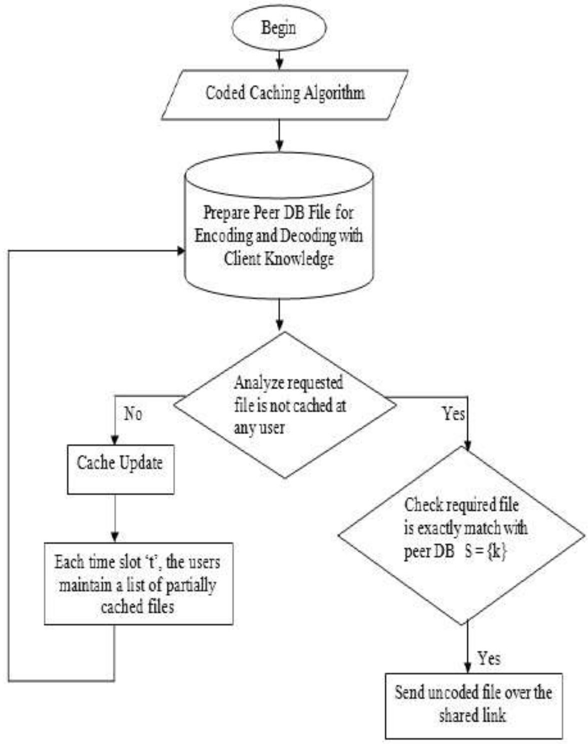

Fig. 2. Flowchart for caching with coded file.

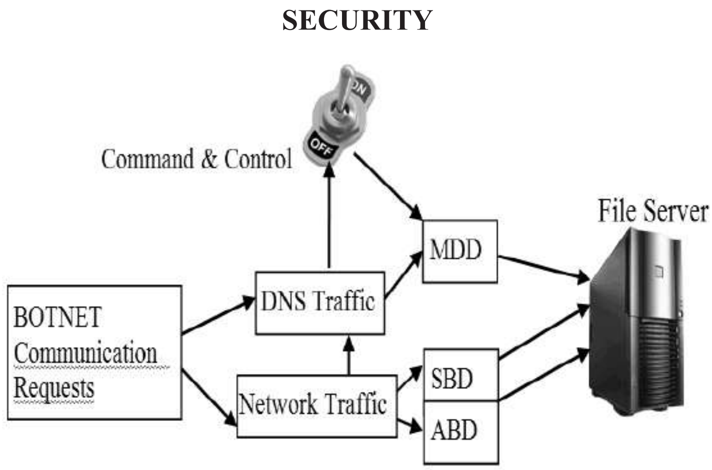

Fig. 3. Commands and control file security server system.

DNS identifies and utilizes command and control server. In this paper, they give framework of system to identify advanced persistent threat (APT), which takes the delicate information. It utilizes malicious domain name service (DNS) examination system for identifying APT malware, as it is difficult to recognize. APT can attack ant viruses passed by firewalls. In the event of a change of Internet Protocol communication address of command and control server into the malware binary, then it causes harm to the system as it conceals the attack of the invader. As APT malware relies on DNS for assault, it is to be expelled. It is altogether dissimilar from bots and worms. APT pedals the machines and takes private information, rather than dispatching DOS physical attack. There are two outlines of BOTNET correspondence, that is, by means of resolver, which utilizes DNS resolver, and by means of non-resolver, which does not. The principle objective is to identify DNS based BOTNET correspondence or communication. Figure 3 shows the structural design of command and control file security server system. It is comprised of four primary components:

BOTNET correspondence requests: In the network system, BOTNET is placed and it is boundary to trace the inspiring and outspring traffic created with network (Chae et al, 2016).

Malicious DNS detector (MDD): It is identifying doubtful APT malware command and control server domain for the "network traffic commentator'" of system.

Network traffic commentator: It is comprised of signature based detector (SBD) and anomaly based detector (ABD). The SBD has characterized C&C correspondence traffic signature for identifying malware identified by the framework of system. The ABD identifies abnormal practices including protocol oddity, measurable oddity, and application oddity (Janakiraman & Suthir, 2014; Suthir & Janakiraman, 2015).

Repute file server: It reviews whether the server possessing the Internet Protocol communication address is tainted, by utilizing malicious domain name service and network traffic characteristic vectors in concert.

PARAMETER PLACING FOR FAST FILE SHARING WITH NETWORK CODING INFLUENCE

Herewith, how swift file sharing is done using the network coding design and with the proposed SNT algorithm and DCS protocols design work in the network is examined. The parameter places for all modernization processes are stated below.

Table 2. Network environment parameter placing.

Connectivity Object	Ranges
Network Terminal Number	50 Nodes
Network Channel Type	Ethernet LAN Connection (data rate = 10 Mbps)
Peer Upload Links enablement	24 Channels
Peer Download Links enablement	24 Channels
Peer files sending Rate	24*24 Kilo Bit
File object	**Ranges**
Files per Transaction	1 (One File per Transaction)
File Size	2 GB
File Churn size	24 (85.3 MB)
Peer File Encode Rate	26.4 MB
Peer File Decode Rate	78.2 MB

PERFORMANCE ANALYSIS

There are two peer-level properties including content distribution and peer behavior. The simulation results of the proposed algorithm and protocols in file transactions are as follows:

i. In network coding technique, the time taken to complete a file transfer with finding is 5 sec and 72 sec for transaction. Hence, the total computation time is 77 sec.

ii. Next, in SNT algorithm, the time taken to complete a file transaction with finding is 5 sec and 54 sec for transaction. Hence, the total computation time of this algorithm is 59 sec.

iii. By using the first protocol 1D called hasty file searching and deduplication protocol, the time taken for a file transaction with finding is 2.5 sec and 54 sec for transaction. The total completion time is 56.5 sec.

iv. By using the second protocol 1C called caching, the time taken to complete the file transfer with finding is 2.5 sec and 50 sec for transaction. Thus, the total completion time is 52.5 sec.

v. By using the last protocol 1S called security with command and control, the time taken to complete the transaction with finding time is 2.5 sec and 50 sec for transaction. Hence, the completion time is 52.5 sec.

vi. Finally, with the use of 1 and 3 (SNT and DCS), finding time is 2.5 sec and transaction is 50 sec. Thus, the total time is 52. 5 sec.

Unlike different strategies that utilize broadcasting to cluster nodes, the concept leverages the SNT algorithm with DCS protocols giving fast file sharing over the network with the above parameters. These techniques are extensively analyzed and affirmed to improve the overall performance of fast file sharing system in a network. The SNT is the most important file transfer technique. A SNT algorithm is used for powerful peer companion in both cluster and sub-cluster. The convention of file searching and deduplication protocol has been obtained for fast file searching and determining the reputation of files.

Table 3. Simulation results of the proposed algorithm and protocols used in file transaction

Techniques Used	**Finding (F), Transaction (T), Completion Time (C) in Seconds**
With use of Network Coding (NC)	F= 5, T= 72, C= 77
With use of SNT Algorithm	F= 5, T= 54, C= 59
With use of Protocol 1 (D) *(Hasty File Searching & Deduplication)*	F=2.5, T= 54, C= 56.5
With use of Protocol 2 (C) *(Caching with Coded)*	F=2.5, T= 48, C= 50.5
With use of Protocol 3 (S) *(Security with Command & Control)*	F=2.5, T= 50, C= 52.5
With use of SNT & DCS	F=2.5, T= 50, C= 52.5

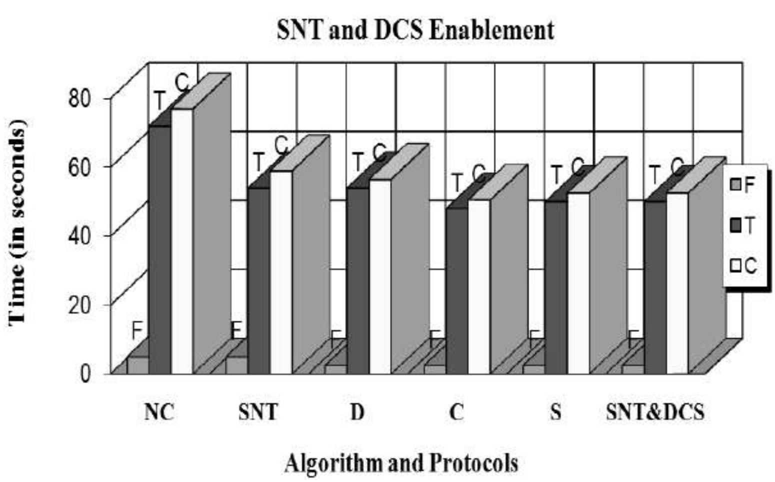

Fig. 4. The effectiveness of enabling the proposed algorithm and protocols.

The time complexity of the proposed techniques is as follows: network coding is 77; the time complexity of SNT technique is 59; hasty file searching and deduplication (protocol 1) with time complexity is 56.5; the time complexity of caching technique (protocol 2) is 50.5; and the time complexity of security (protocol 3) is 52.5.

Table 4. Comparison of various techniques

S. No	Reference	Existing Techniques	Applications
1	Javanmardi, S., Shojafar, M., Shariat-madarI, S. & Ahrabi, S. S., 2014	P2P Grid Semantic Web OS	Better search expressiveness. Improved resource utilization. Better accuracy and scalability.
2	Javanmardi, S., Shojafar, M., Shariat-madarI, S. & Ahrabi, S. S., 2015	Fuzzy Reputation Trust Based Technique	Better computational capability. High ranking accuracy.
3	Miltchev, S., Prevelakis, V., Ioanni-dis, S., Keromytis, A. D. & Smith, J. M., 2003	Credential based technique	Flexibility and scalability. Straightforward to implement.
4	Ramaswamy, R. & Liu, L., 2003	Simulation technique	Increase the system lifetime.

CONCLUSION AND FUTURE WORK

This paper has presented the SNT algorithm and DCS protocol for the file sharing processes in networks with the influence of network coding. The study is all around adjusted algorithm and protocol for expansive size of coefficient lattices; particularly, when the quantity of nodes and solicitations turns out to be immense, execution upgrading turns out to be observable. The content sharing system framework is measured with a single server associated in the course of mutual shared connection to clients. A network caching issue has been identified, in which clients request a succession of data/files from a place of prevalent data/files at every moment vacancy. Moreover, an SNT algorithm was proposed for powerful peer companion in cluster and sub-cluster. Deduplication and file searching protocol convention have been taken for fast file seeking and evacuating the copy of files. Then network caching with coded protocol convention is considered, which exhibits incredible favourable position over the routine cache memory with encoded and decoded plot. Lastly, the command and control security protocol has been dealt with, which provides security over file transactions. This outcome implies that these algorithms and protocols together provide an overall performance of network file transaction that has accomplished 14 percent speedup of file sharing compared to previously proposed concepts, methods, and algorithms. These outcomes demonstrate that the versatility of SNT algorithm and DCS protocols can be received in the future for many centre framework systems. As an upcoming job, we determine research taking place for the completion problem of substantial functions such as peer-to-peer network file sharing projects. In addition, we likewise need to catch this existent world network holdup into thought in favour of the exact presentation design.

REFERENCES

Ahlswede, R., Cai, N., Li, S.-Y.R. & Yeung, R.W. 2000. Network Information Flow, IEEE Trans. Information Theory, Vol. 46(4): pp. 1204 - 1216 .

Asano, T., Ranjan, D., Roos, T., Welzl, E. & Widmaier, P. 1997. Space filling curves and their use in geometric data structure," Theoretical Computer Science, vol. 181, no. 1, pp.3–15.

Chae, C., Shin, J., Choi, K., Kim, K. & Choi1, K. 2016. A privacy data leakage prevention method in P2P networks. in Peer-to-Peer Netw. Appl. (2016) 9:508–519.

Chou, P., Wu, Y. & Jain, K. 2004. Practical Network Coding. Proc. 51st Allerton Conf.Comm., Control and Computing.

Darlagiannis, V., Heckmann, O. & Steinmetz, R. 2006. Peer-to-Peer applications beyond file sharing: overlay network requirements and solutions. In Elektrotechnik & Informationstechnik, 123(6): 242–250.

Ding, C. H., Nutanong, S. & Buyya, R. 2005. Peer-to-Peer Networks for Content Sharing. Peer-to-peer computing journal, 1 - 32.

Gkantsidis, C. & Rodriguez, P.R. 2005. Network Coding for Large Scale Content Distribution. Proc. IEEE INFOCOM '05, Vol. 4, pp. 2235 - 2245.

Ho, T., Me´dard, M., Koetter, R., Karger, D.R., Effros, M., Shi, J. & Leong, B. 2006. A Random Linear Network Coding Approach to Multicast. IEEE Trans. Information Theory, Vol. 52, No. 10: 4413 - 4430.

Janakiraman, S. & Suthir, S. 2014. Fashionable and Effectual Collective Area Setup in Node-to- Node Message. In International Journal of Software and Web Sciences (IJSWS) with International Association of Scientific Innovation and Research (IASIR) ISSN (Print): 2279 - 0063 ISSN (Online): 2279 - 0071, Issue. 7, Vol. 1, pp.08 - 12.

Javanmardi, S., Shojafar, M., ShariatmadarI, S. & Ahrabi, S. S. 2014. PGSW-OS: a novel approach for resource management in a semantic web operating system based on a P2P grid architecture. The journal of supercomputing, 69 (2): 955 - 975.

Javanmardi, S., Shojafar, M., ShariatmadarI, S. & Ahrabi, S. S. 2015. FR trust: a fuzzy reputation–based model for trust management in semantic P2P grids. International Journal of Grid and utility computing, 6(1).

Katti, S., Rahul, H., Hu, W., Katabi, D., Me´dard, M. & Crowcroft, J. 2008. XORs in the Air Practical Wireless Network Coding. IEEE/ ACM Trans. Networking, 16(3): 497 - 510.

Leu, J., Yu, M. & Yueh, H. 2015. Improving Network Coding Based File Sharing for Unstructured Peer-to-Peer Networks. In Journal of Network System Management 23:803–829.

Li, J., Chen, X., Huang, X., Tang, S., Xiang, S. 2015. Secure Distributed Deduplication Systems with Improved Reliability. In IEEE Transactions on Computers, vol. 64, no. 12.

Lun, D.S, Ratnakar, N, Koetter, R, Karger, D.R, Ho, T, Ahmed, E. & Zhao, F. 2006. Minimum- Cost Multicast over Coded Packet Networks. IEEE/ACM Trans. Networking, vol. 14, no. 6, pp. 2608 - 2623.

Maddah-Ali, M. A. 2015. Decentralized Coded Caching Attains Order-Optimal Memory-Rate Tradeoff. In IEEE/ACM Transactions on Networking, vol. 23, no. 4.

Miltchev, S., Prevelakis, V., Ioannidis, S., Keromytis, A. D. & Smith, J. M. 2003. Secure and Flexible Global File Sharing. USENIX annual technical conference.

Park, K., Park, J. & Ro, W. W., 2011. On Improving Parallelized Network Coding with Dynamic Partitioning. In IEEE Transactions on Parallel and Distributed Systems, Vol. 21.

Pedarsani, R. & Maddah-Ali, M. A. 2016. Online Coded Caching. In IEEE/ACM transactions on Networking, Vol. 24, No. 2.

Ramaswamy, R. & Liu, L. 2003. Free Riding: A New Challenge to Peer-to-Peer File Sharing Systems. In Proceedings of the Hawaii International Conference on Systems Science.

Ratnasamy, S., Handley, M., Karp, R. & Shenker, S. 2002. Topologically-aware overlay construction and server selection. In Proc. IEEE INFOCOM, pp. 1190 - 1199.

Shen, H. & Liu, G. 2015. A Proximity-Aware Interest-Clustered P2P File Sharing System. In IEEE Transactions on Parallel and Distributed Systems, Vol. 26, No. 6.

Shen, H. & Xu, C.-Z. 2008. Hash-based proximity clustering for efficient load balancing in heterogeneous DHT networks. J. Parallel Distrib. Comput., vol. 68, pp. 686 - 702.

Shojafar, M. & Jemal, H. Abawajy, Pooranian, Z. & Abraham, A. 2015. An efficient and distributed file search in unstructured peer-to-peer network's. In Peer-to-Peer Netw. Appl. 8:120 - 136.

Suthir S & Janakiraman S, S. 2013. Contemporary and efficient shared area network in Peer-to- Peer Communication. In IEEE Xplore - International Conference on Radar, Communication and Computing (ICRCC), pp.38 - 42, ISBN: 978 - 1467 327 - 589.

Suthir S & Janakiraman S, 2014. A Hybrid Network Model for Increasing Bandwidth Using Nano Communication. In International Journal of Scientific & Engineering Research, ISSN 2229 - 5518, Vol. 5, Issue. 3, pp. 222 - 225.

Suthir S & Janakiraman S, 2015. A Malicious Discovery Exposure Scheme Towards Resourceful Reliance Authority in Delay Forbearing Network. In International Journal of Science, Technology & Management, ISSN (online): 2394 - 1537, Vol.04, Issue. 03, pp.228 - 233.

Suthir S & Janakiraman S, 2016. A Contemporary Network Security Technique using Smokescreen SSL in Huddle Network Server. In IEEE Xplore - International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB16), pp.673 - 676.

Wang, N. & Ansari, N. 2011. Downloader-Initiated Random Linear Network Coding for Peer-to- Peer File Sharing. In IEEE Systems Journal, Vol. 5, No. 1.

Wang, M. & Li, B. 2007. Lava: A Reality Check of Network Coding in Peer-to-Peer Live Streaming. Proc. IEEE INFOCOM '07, pp. 1082 - 1090.

Xu, Z., Mahalingam, M. & Karlson, M. 2003. Turning heterogeneity into an advantage in overlay routing. In Proc. INFOCOM, pp. 1499 - 1509.

Yang, M. & Yang, Y. 2014. Applying Network Coding to Peer-to-Peer File Sharing. In IEEE Transactions on Computers, 63(8).

Zhao, J., Feng, N., Liu, X., Bai, W. & Maier, M. 2014. Implementing Network Coding in Service Interoperability Ethernet Passive Optical Network (IEEE P1904.1 SIEPON). In J. OPT. Commun. Network, Vol. 6, No. 5.

Zhao, G. 2015. Detecting APT Malware Infections Based on Malicious DNS and Traffic Analysis. In IEEE ACCESS, vol. 3, pp.1132 - 1142.

Submitted: 17/05/2017
Revised : 30/09/2017
Accepted : 30/10/2017

Journal of Engg. Research **Vol. 6 No. (3) September 2018 pp.** *69*

اندماج خوارزمية SNT وبروتوكولات DCS لتعزيز مشاركة الملفات المعاصرة مع تأثير تشفير الشبكة

سوثير اس وجاناكيرامان اس
جامعة مانونمانيام ساندرانار، الجامعة التكنولوجية، تيرونلفلي
تكنولوجيا المعلومات، جامعة بونديشيري

الخلاصة

تشفير الشبكة في الوقت الحالي اجتذب اهتماماً رائعاً لتحسين انتاجيتها. في هذا البحث، تم تقديم خوارزمية SNT وبروتوكولات إلغاء التكرار، التخزين المؤقت والحماية (DCS) لتحسين أداء المشاركة البينية أثناء عملية تبادل الملفات السريعة مع تشفير الشبكة الذي يمكن أن يصل إلى 10٪ إلى 15٪ في معدل النقل بالمقارنة مع ناتج شبكة P2P المتعددة البث الأخرى. في البداية اقترحنا العثور على أقرب نظير على الشبكة باستخدام خوارزمية SNT. ثانياً، تم تحليل دوال البروتوكولات المستخدمة في النهج المقترح، حيث تم في البروتوكول الأول إلغاء البيانات المكررة لنسخ طبق الأصل من الملفات للحفاظ على مساحة تخزين قاعدة البيانات بشكل فعال. ويعمل البروتوكول الثاني وهو التخزين المؤقت على العثور على عقدة طلب ملف في تدفق البيانات. والبروتوكول الأخير هو إعطاء الأوامر والتحكم لتوفير الحماية والتحقق من عنوان IP وعدم تطابق عقد الإرسال الصاعدة والهابطة.