Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 79 (2016) 53 – 60

Procedia Computer Science

7th International Conference on Communication, Computing and Virtualization 2016

Content-based file sharing in peer-to-peer networks using threshold

Amol Bhagat[a], Radhika Chaudhari[a], Kiran Dongre[a]

[a]Innovation and Entrepreneurship Development Center, Prof Ram Meghe College of Engineering and Management. Badnera, Amravati, 444701, India

Abstract

In content based file sharing peer-to-peer (P2P) [1] network model nodes share files directly with each other without a centralized server. In such a file sharing system, nodes meet and exchange requests and files in the format of text, short videos, and voice clips in different interest categories. Content is various and large file sharing such as the multimedia content is required with the rapid development of the wireless communication technology. File sharing can also mean having an allocated amount of personal file storage in a common file system. A P2P content based file sharing system, for efficient file searching, threshold takes advantage of node mobility by designating stable nodes, which have the most frequent contact with community members, as community coordinators for intra community searching, and highly mobile nodes that visit other communities frequently as community ambassadors for intercommunity searching. The large file sharing needs more stable end to end path and long transmission time. Last but not least, more relationship between nodes will be used to promote the file sharing process. Content based file sharing is helpful for taking certain decisions during file transmission. These decisions will benefit in proper utilization of network resources. In this paper content-based file sharing scheme using threshold is proposed. The user's interest is determined by the proposed scheme before searching and sharing the files in the peer-to-peer network. The resources in the network are utilized as per the contents of the files to be shared. The performance evaluation show that proposed system significantly lowers transmission cost and improves file sharing success rate compared to current methods.

Peer-review under responsibility of the Organizing Committee of ICCCV 2016

Keywords: Content-based file sharing; interest extracton; interest oriented file sharing; peer-to-peer network; threshold based sharing;

1. Introduction

Identifying file types (ASP, JPG, EXE, etc.) is a non-trivial task that is required in many computer applications. File types are usually identified by the file extensions [1] or the magic numbers in the file header [2]. However, these methods can easily be deceived by changing the file extension or magic number. Therefore, especially in the presence of adversaries, a more reliable solution is needed. A typical scenario is a course material (e.g., course slides, review sheets, assignments) sharing system in a school campus. Such a scenario ensures for the most that

Peer-review under responsibility of the Organizing Committee of ICCCV 2016
doi:10.1016/j.procs.2016.03.008

nodes sharing the same interests (i.e., math), carry corresponding files (i.e., math files), and meet regularly (i.e., attending math classes). Analysing file contents to find distinguishable patterns between different file types is available alternative but it is not wide spread, because it is inefficient and time-consuming. Existing approaches generate the byte-frequency distribution of a file and use it for classification with statistical or data mining techniques. The calculations required for the distribution may be time consuming because it scales with file size. Also, classifying file types with distributions may require a large memory space and computation time, since the computation time exponentially increases with the number of sequences of n-bytes (n-gram).A suitable content sharing paradigm must minimize the consumption of network resources and must divide the burden of sharing data equally among the set of nodes by thinking about the topology of the network and giving enough incentives for fair sharing. In general, a P2P file sharing system mainly consists two parts: search algorithm and a file transfer protocol. File transfer protocol is responsible to downloading files by use of TCP connection. While search algorithm is responsible for transmitting query messages and searching results. And search algorithm, which has more relationship with efficiency, attracts more attention.

The values of the thresholds used in the file sharing process are determined by many factors such as number of nodes, number of interests, and type of files. This is based on seating up a threshold value so that the traffic flows exceeding the threshold will be classified for taking certain decision. Two encountered nodes exchange their interest vectors and community vectors. In the role assignment phase, nodes broadcast their degree centrality within their communities for coordinator selection. When the coordinator is selected, the coordinator ID is also broadcasted to all nodes in the community. Then, each node reports its contact frequencies with foreign communities to the coordinator for ambassador selection. Besides, when a node meets a coordinator of its community, the node also sends its updated node vector to the coordinator to update the community vector and retrieves the updated community vector from the coordinator. When an ambassador meets the coordinator of its community, it reports the community vectors of foreign communities to the coordinator. After above information exchange, two encountered nodes exchange their node vectors and history vectors for packet routing. Each node checks packets in it sequentially to decide which packets should be forwarded to the other node based on the file searching algorithm. Further, when network turns to be stable, the frequency of information exchange for community construction and node role assignment can be reduced to save costs. In this paper the performance parameters used in different content based file sharing in peer-to-peer networks are evaluated. Existing system fails due to the large overhead involve in the transmission of files with distinct contents. This disadvantage of existing system has been solved by content based file sharing in peer-to-peer network using threshold. Since the file sharing client may share file with identified parameters with appropriate threshold value according to the content of file.

2. Related Work

With the everyday increasing importance of privacy, security, and wise use of computational resources, the corresponding technologies are increasingly being faced with the problem of file type detection. True identification of computer file types is a difficult task especially when dealing with suspicious goals. Computers are dealing with the huge number of file formats that are transmitting between the insider and outsider networks. Without the true file type detection, the security will not be achievable. File type detection has the most usage and importance in the proper functionality of operating systems, firewalls, intrusion detection systems, anti-viruses, filters, steganalysis, computer forensics, and applications dealing with the file type classification [4]. Although there are many applications dealing with file type detection, they have very limited methods for determining the true type of files. The newest method of file type detection is based on the file contents. McDaniel and Heydari published the first paper on content-based file type detection [5, 6]. They introduced three introductory algorithms for the content-based file type detection: Byte frequency Analysis (BFA), Byte Frequency Cross-correlation (BFC), and File Header/Trailer (FHT) analysis. They deployed such algorithms on the whole content of sample files to produce a fingerprint of each file type. Li et al. **[7]** performed some improvements on the McDaniel and Heydari's method to improve its accuracy. Their improvements was using multi-centroids for some file types, truncating the sample files from their beginning instead of using all the file contents, using KMeans algorithm under Manhattan distance to produce the file prints and deploying Mahalanob is distance for comparison between the unknown samples and file prints. Dunham et al. [8] deployed neural networks to classify file types of stream ciphers in depth, i.e. the files encrypted with the same key. They used byte frequency, byte frequency of autocorrelation, and 32 bytes of header as the selected features of their test samples. Karresand and Shahmehri [9] used the mean and standard deviation of Byte Frequency Distribution (BFD) to model the file types. Their method is based on data fragments of files and does not need any metadata. Zhang et al. [10] used the BFD in conjunction with a simple Manhattan distance comparison to detect whether the examined file is executable or not. A new content-based file type detection method is introduced that deploys the Principal Component Analysis (PCA) and unsupervised neural networks for the automatic feature extraction.

Content-based file-type identification schemes often use byte-frequency distribution as a feature and use statistical and data mining techniques to classify file types. Since those schemes use the entire file content to obtain byte-frequency distribution and use all possible byte patterns in file classification, they are inefficient and time-consuming. The system proposes two techniques to reduce the classification time. The first method is a feature selection technique, which uses a subset of highly-occurring byte patterns in building the representative model of a file type and classifying files [11]. To evaluate its effectiveness, we applied it to the six most popular classification algorithms (i.e. neural network, linear discriminate analysis, K-means, K-nearest neighbor, decision tree, and support vector machine). On average, the K-nearest neighbor method achieved the optimum accuracy of 90% using only 40% of byte patterns; this reduces 55% of computation time. The second method is the content sampling technique, which uses a small portion of a file to obtain its byte-frequency distribution. It is effective for large size files where a relatively small sample can generate the representative byte frequency distribution. For instance, it reduces the sampling size of MP3 _les from 5MB to 400KB (without compromising the accuracy). This is a 15 fold size reduction.

McDaniel and Heydari [12] introduced three algorithms to analyze file content and identify file types. The byte-frequency analysis algorithm (BFA) computes the byte-frequency distribution of different files and generates "fingerprint" of each file type by averaging the byte-frequency distribution of their respective files. To obtain another characterizing factor, they also calculate the correlation strength as by taking the difference between the same byte in different files. As the difference becomes smaller, the correlation strength approaches to 1, and vice versa. The byte-frequency cross correlation algorithm finds the correlations between all byte pairs. It calculates the average frequencies of all byte pairs and the correlation strength in a similar manner to the BFA algorithm. The file header/trailer algorithm uses the byte-patterns of the file headers and trailers that appear in a fixed location at the beginning and end of a file, respectively. It maintains an array of size 256 for each location and the array entry corresponding to the byte is filled with the correlation strength of 1. It constructs the fingerprint by averaging the correlation strength of each file. In these algorithms, they compare the file with all the generated fingerprints in order to identify its file type.

Repantis and Kalogeraki propose a file sharing mechanism in [13]. Nodes use the Bloom filter to build content synopses of their data and then disseminate them adaptively to the most appropriate nodes. Additionally, the methods cannot guarantee the successful searching of file, especially when the routes expire due to node mobility. Li and Wu propose a Mobile community-based Pub/Sub scheme (MOPS) [14] to implement the content-based service. MOPS seek to utilize long-term neighboring relationship between nodes to construct the community in the DTNs. And the community is defined as a clique of nodes where any neighboring relationship is stronger than an adjustable threshold. In addition, MOPS adapts the pub/sub paradigm which integrates push and pull, to determine the interface known as the push-pull boundary and deploy brokers to bridge the boundary. The brokers propagate interests and collect events by utilizing a unique weighted scheme. Extensive real- and synthetic-trace-driven simulation is made and the results are presented to support the effectiveness of MOPS.

Palazzi and Bujari present a special purpose system for searching and transferring files which based on an application layer overlay network and port a DTN type solution into an infrastructure-less environment like MANETs [15]. They leverage peer mobility to reach data in other disconnected networks which implements a store-delegate-and-forward asynchronous communication model to delegate unaccomplished file download or query tasks to special peers. To improve data transmission performance while reducing communication overhead, they select these special peers by the expectation of encountering them again in future and assign them different download starting point on the file. Qureshi et al. present an adaptive protocol to implement a Mobile Social Network based on P2P content driven communication when end-to-end connectivity is not possible [16]. The proposed protocol considers the information about user’s interests, content based data storing and forwarding, and host mobility in a disconnected and delay tolerant MANETs. The authors define a three layer stack in the protocol. The top layer supports the user interface which works as an application layer. Then the middle layer provides support for content driven data dissemination in the form of documents and messages. And the third layer is responsible for data forwarding to distant nodes in a multi-hop manner. In order to unicasting messages from point to a specific point in the network, the protocol considers using Ad hoc on-demand Distance Vector (AODV).

Table 1. Summary of content-based file sharing approaches.

Methodology	Purpose	Models/ Concept	Comparison Carried out	Evaluation Parameters	Purpose
Social-based method	Improve file searching success rate	Node role assignment	Peer-to-Peer and MANET	MOPS, PDI+DIS, CacheDTN, Podnet and Epidemic	Improve file searching success rate
Broadcast-based and DHT-based	cross-layer protocols perform better than simply overlaying peer-to-peer searching protocol on mobile ad hoc networks.	Cross-layer protocol	Routing complexity, scalability, complexity, maintenance complexity, energy efficiency, the shortest path and cross-layer protocol.	Complexity	cross-layer protocols perform better than simply overlaying peer-to-peer searching protocol on mobile ad hoc networks.
Hashline and tree-shape topology	Able to find the location of a file in a WANET	Transport and Network layer	Node-Join, Network-Join, Access2P-Node, Access2F-Node, Insert, Delete, Recover, Leave	Reliability and Availability	Able to find the location of a file in a WANET
----------	A new structure that captures common user interests in data—the data-sharing graph—and justify its utility	Network type	Distribution of Weights	data-sharing graph and its small world properties.	A new structure that captures common user interests in data—the data-sharing graph— and justify its utility
7DS (“Seven Degrees of Separation”)	To accelerate the data availability and enhance the dissemination and discovery of information when hosts face changes in the bandwidth availability.	-------	Peer-to-Peer (P-P) Server-to-Client(S-C)	wireless coverage range, 7DS host density, querying mechanism, energy conservation	To accelerate the data availability and enhance the dissemination and discovery of information when hosts face changes in the bandwidth availability.
iTrust over SMS,	Mobile search and retrieval	mobile device	----------	iTrust over SMS using mobile phones in large real-world social networks	Mobile search and retrieval

ContentPlace [14] defines social relationship based communities and it assumes that users belong to several different social communities, and automatically learns the time spent by them in each community, which types of data users of each community are interested in, and how spread in communities the data are. By using information, each node calculates a utility value for each encountered object regarding its connected communities and caches these objects in a highest utility value first manner. Specifically, each node, upon making contact with another peer, evaluates the utility of the data the peer is carrying and decides which data to fetch from the peer, in order to maximize the total utility of the data in its own buffer. Content Place considers five policies to evaluate the social weight: Most Frequently Visited (MFV), Most Likely Next (MLN), Future (F), Present (P), Uniform social (US) and the simulation provide best results in the Most Likely Next and Future policies. Tchakarov and Vaidya present a content location service the Geography-based Content Location Protocol (GCLP), which takes physical location information into consideration in order to provide an efficient content location service to nodes in an ad hoc network [15]. GCLP tries to allow each client to find a nearby server by use of the geographical distance as the distance metric (not number of hops). In GCLP, nodes make use of geographic information to advertise content periodically which are hosting to nodes along several geographical directions.

A special-purpose on-demand file searching and transferring algorithm based on an application layer overlay network is proposed in [12]. The algorithm transparently aggregates query results from other peers to eliminate redundant routing paths. A peer-to-peer file sharing system that is running on Internet may find a desired file at a member node, which is identified by a unique ID. This can be achieved by using centralized or distributed indices that maps the name of the file to the member node's IP address through which the node can be reached. After knowing the IP address of the node from where a file can be downloaded, the network layer of the Internet (IP) would handle all intermediate steps and forwarding needed in order to reach to the node and to perform the download. However, this is not possible on a WANET that does not run an ad-hoc routing algorithm. A WANET may be composed of heterogeneous mobile systems in which a standard routing algorithm is not supported at all nodes. Therefore, to support peer-to-peer file sharing in a WANET, we believe that a peer-to-peer system should also provide routing functionality besides providing lookup functionality. In this way, the peer-to-peer system should be able to determine both from where and how to obtain a file.

Mobile ad hoc networks are a mobile, dynamically and self-organizing wireless networks in absence of a fixed infrastructure, which is usually used in emergency environment such as disaster recovery, military battlefields etc. Recent years, the personal mobile devices such as smart phone, PDA, ipad are increasing rapidly. The mobile user would like to generate personal content, store useful information, search content from Internet or other mobile devices and share content with their friends all in handheld mobile platforms. However, the limited communication capacity of the base station is not easy to satisfy the big requirement. MANETs [16] consisted by the mobile devices which are sufficient with wireless technologies such as WIFI or Bluetooth, are effective and important supplement to the application between mobile users. The mobile devices bring new application scenario to MANETs. File sharing in MANETs attracts more and more researchers' attention. A lot of proposal discusses the P2P file sharing on MANETs and we classify them into four kinds according to their varying searching principle: Flooding-based method, advertisement-based method, cache/replication-based methods and social-based method. The comparison of all the above discussed strategies is shown in table 1.

3. Proposed Approach for Content-based File Sharing

The proposed content-based file sharing in peer-to-peer networks using threshold is shown in Fig 1. File sharing client allows a number of people to use the same file or files by some combination of being able to read or view it, write to or modify it, copy it, or print it. Typically, a file sharing system has one or more administrators. File sharing clients may all have the same or may have different levels of access privilege. This file sharing client generated request to share file. In request analysis, the request is analyzed to see whether it is authorized or not. If it is authorized then process will be continued otherwise it is stop there. The sample files are collected from different sources to ensure that the sample files of a file type are not generated by one source. Moreover, the other files are collected from the internet using a general search on Google. For example, .txt file was searched using option "filetype:txt". The image files such as GIF and JPG are also obtained from photo sharing websites such as Picassa of Google and flickr etc. The MP3 files are collected from different random sources mostly from the publicly available FTP servers for movies and personal computers. Identify physical location of analyzed request according to their content, size, type etc. According to the physical location of the file extract the contents from file. There are two types of contents low level and high level. As per the file contents determine transmission requirement of the file. Last share file with identified parameters which is help to decide the threshold value. Keep all shared files in one database and determine user/nodes interest.

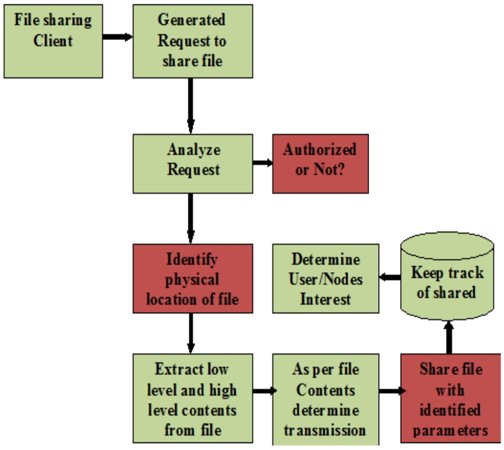

Fig. 1. Proposed system architecture for content-based file sharing using threshold.

It is identified that users usually have a few file categories that they query for files frequently in a file sharing system. To derive its interest a node infers keyword from each of its files using the document clustering technique [3]. A node derives a file vector for each of its file from its metadata. Each file forms an individual group initially. The two most similar file groups are merging in each step. This process repeats until the similarity between any two groups is below a threshold. The similarity between two groups is calculated based on their interest vectors introduced below. Consequently, a file is classified to only one interest group and there is no overlap among groups. Nodes with multiple interests belong to multiple communities. The community construction can easily be conducted in a centralized manner by collecting node interests and contact frequencies from all nodes to a central node. Also nodes in a group admit new members distributive. As a result, nodes in a group may not have very similar interests or high contact frequencies. There are two solutions to alleviate this problem. First, set an initial period for newly joined nodes in which they accumulate contact frequencies with others. Then, when a node starts to join in communities, its meeting frequencies with others are relatively stable, which provides more accurate measurement for determining the communities to join in. Second, use group member pruning. Existing community members can have a second round voting to confirm the eligibility of new community members. The values of the thresholds used in the community construction process.

Increase the node mobility for efficient file searching. A community coordinator is an important and popular node in the community. It keeps indexes of all files in its community. Each community has one ambassador for each known foreign community, which serves as the bridge to the community. The number of ambassadors and coordinators can be adjusted based on the network size and workload to avoid overloading these nodes. Community coordinator node selection: A stable node that has tight contact frequency with other community members as the community coordinator. Community ambassador node selection: An ambassador is used to bridge the coordinator in its home community and a foreign community. The interest-oriented file searching scheme has two steps: intracommunity and intercommunity searching. A node first searches files in its home community. If the coordinator finds that the home community cannot satisfy a request, it launches the intercommunity searching and forwards the request to an ambassador that will travel to the foreign community that matches the request's interest. A request is deleted when its time-to-live (TTL) expires. During the search, a node sends a message to another node using the interest-oriented routing algorithm, in which a message is always forwarded to the node that is likely to hold or to meet the queried keywords. The retrieved file is routed along the search path.

4. Experimental Evaluation

In this work inverse document frequency is utilized for analyzing the contents of the files in peer-to-peer network. The basic of the inverse document frequency theory is: Given a query Q, containing keywords the BM25 score of a document D is calculated as,

$$\textbf{Score (D, Q)} = \sum_{i=1}^{n} IDF(qi) \, . \, \frac{f(qi,\ D)\,.\,(k1+1)}{f(qi,\ D)+k1\,.(1-b+b\frac{|D|}{avgdl})}$$

where f(qi, D) is qi's term frequency in the document D, | D | is the length of the document D in words, and avgdl is the average document length in the text collection from which documents are drawn. k1 and b are free parameters, usually chosen as k1 = 2.0 and b = 0.75. IDF (qi) is the IDF (inverse document frequency) weight of the query term qi. IDF is usually computed as: tf-idft = log (N/n(q)) where N is number of documents in collection and N(q) is the number of documents containing q.

Suppose a query term q appears in n (q) documents. Then a randomly picked document D will contain the term with probability (where N is again the set of documents in the collection). This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tfidf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. The weight of a word in a document is higher if it occurs many times within a small number of documents (thus lending high discriminating power to those documents) the weight of a word is lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal) the weight of a word is lowest when the term occurs in virtually all documents.

The probability of picking a document containing two terms has nothing to do with the intrinsic relevance of those two terms inside a document thus to the relevance of the document to other documents in collection, related to the query terms. The computed score is just a some kind of a probability score that has nothing to do with the relevance of a document among others, related to a query. Misleading probability for relevance is the first big error. The formula was extended to a database with virtually an infinite number of documents. It also refers to words related to other words, not to documents related to other documents. Each document is a set of concepts. The proposed approach try to match the concepts users look for with the concepts inside documents.

Table 2. Searched files in the created P2P environment using content-based file search using threshold.

Searched Keyword	Files Returned as Search Result	Time Required for Searching (Milliseconds)	Bandwidth Requirement for File Sharing (Transfer Kbps)
Calendula Image	S8.txt	229	256
	R10.txt		128
	P1.txt		128
	calendula.jpg		512
	S5.txt		256
Bamboosky Image	S2.txt	214	256
	R2.txt		256
	bamboosky.jpg		1024
Autumleaf Image	P2.txt	233	128
	S1.txt		256
	Autumleaf.jpg		512
Wild Life	S6.txt	2345	32
	P2.txt		56
	P5.txt		128
	Wildlife.wmv		2048
	R9.txt		256

In the configured peer-to-peer environment for file sharing, peers request some information about the other peers and the current place of the shared file, then server responds to their request by giving some information about the shared files such as File size, Peer name, File extension and the available peers. Each computer in a network of computers who are interacting together, called Peer. A cloud network, formed in the peer-to-peer environment is a mesh network which has a specified address scope and this scope is related to an IPV6 scope. Peers in a cloud are those ones who can communicate across this scope. There are 2 types of predefined cloud as below: global cloud if a computer connects to the internet, then, it is joined to a global cloud and link local cloud which contains a set of nodes which are connected via LAN, they are working on a link local cloud. Peer Name Resolution Protocol (PNRP) is used to resolve the peers IDs instead of DNS. Each peer beside its ID gets a name is called PeerName. PeerName can be registered either as secured or unsecured names. Securednames are recommended in private networks and securednames are suggested in the Global network (Internet). First of all, the peer needs to become registered in a cloud. A cloud can be registered programmatically or by using netsh command. The available set of files is initially indexed in the created environment and keywords are searched by analysing the concepts. The files containing this keyword are returned as search results and then files can be exchanged or shared as per their network requirements. Table 2 shows some keywords searched in the network environment and files returned with their bandwidth requirement for exchanging or sharing them in the P2P network.

5. Conclusion and Future Scope

In this paper an approach for content-based file sharing in peer-to-peer environment using threshold is proposed for searching the files as per concepts. The searched files are then can be shared in the P2P environment using the identified network requirements. The network requirements are identified as per the contents to be shared. For this purpose threshold value is utilized for knowing the contents type, size, format, etc. The proposed mechanism can be used in any network environment having limited number of resources and maximum number of transmissions. In future the proposed scheme can be extended for identifying the requirements of the network for best effort network services.

References

1. Schutze H. Silverstein C. Projections for Efficient Document Clustering. *Proc. 20th Ann. Int'l ACM Conf. Research and Development in Information Retrieval*, 1997. p. 74-81.
2. Phunchongharn P., Pornnapa S., Achalakul T. File Type Classification for Adaptive Object File System. *IEEE TENCON Conference*, 2006. p. 1-4.
3. McDaniel M., Heydari M. H. Content based file type detection algorithms. *Proceedings of the 36th IEEE Annual Hawaii International Conference on System Science*, 2003.
4. Dunham J. G., Sun M. T., Tseng J. C. R. Classifying File Type of Stream Ciphers in Depth Using Neural Networks. *The 3rd ACS/IEEE International Conference on Computer Systems and Applications*, 2005.
5. Karresand M., Shahmehri N. File Type Identification of Data Fragments by Their Binary Structure. *Proceedings of the IEEE Workshop on Information Assurance*, 2006, p.140-147.
6. Zhang L., White G. B. An Approach to Detect Executable Content for Anomaly Based Network Intrusion Detection. *IEEE Parallel and Distributed Processing Symposium*, 2007, p.1-8.
7. Ahmed I., Lhee K., Shin H., Hong M. Fast file-type identification. *ACM symposium on applied computing*, 2010. p. 1601-1602.
8. Li W., Wang K., Stolfo S. J., Herzog B. Fileprints: Identifying file types by n-gram Analysis. *Proceedings of the 6th IEEE Systems, Man and Cybernetics Information Assurance*, 2005. p. 64-71.
9. McDaniel M., Heydari M. H. Content based file type detection algorithms. *In proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003. vol. 9, p. 332-338.
10. Repantis T., Kalogeraki V. Data dissemination in mobile peer-to-peer networks. *In Proc. of the 6th Int'l Conf. on Mobile Data Management*, 2005. p. 211-219.
11. Chen K., Shen H., Zhang H. Leveraging Social Networks for P2P Content-based File Sharing in Disconnected MANETs. *IEEE Transactions on Mobile Computing*, 2015.
12. Ding Gang, Bhargava Bharat. Peer-to-peer File-sharing over Mobile Adhoc Networks. *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*, 2004.
13. Iamnitchi A., Ripeanu M., Santos-Neto E., Foster I. The Small World of File Sharing. *IEEE Trans. on Parallel and Distributed Systems*, 2011, vol. 22, no. 7, p. 1120-1134.
14. Maria Papadopouli and Henning Schulzrinne. A performance analysis of 7ds a peer-to-peer data dissemination and prefetching tool for mobile users. Advances in wired and wireless communications, IEEE Sarn off Symposium Digest, March 2001. Best student paper & poster award.
15. Bhagat Amol P., Mehare Jayant, Malve Pravin. Estimation of Resource usage in Peer to Peer Network. *International Journal of Computer Applications*, 2012. p. 21-24.
16. Bhagat A. P., Pawade P. P., Jadhav R. D. A Novel Efficient Approximate Query Processing In P2P Network. *Int. J. Advanced Networking and Applications*, 2012. p. 23-28.
17. Bhagat, A. P., Harle, B. R. Materialized view management in peer to peer environment. *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, 2011. p. 480-484.