Resilient and Secure File Dispersal in a Mobile P2P System

Khalid Ashraf[1], Rachid Anane[2] and Nick Blundell[1]

[1]*School of Computer Science, University of Birmingham*
{k.ashraf, n.blundell}@cs.bham.ac.uk
[2]*Faculty of Engineering and Computing, Coventry University*
r.anane@coventry.ac.uk

Abstract

The convergence of mobile networks and P2P systems has led to the deployment of a range of schemes for file storage over a network. This paper presents an approach to resilient file management and network storage through the design and implementation of a file dispersal system for a mobile P2P network, based on distributed hash tables (DHT). Inherent issues of security, reliability and efficiency are addressed by integrating two forms of data redundancy, pure replication and erasure coding. The functionality of the system and its configuration were enhanced at different levels by a variety of security-related techniques. Experimental results on the different stages of the file dispersal process were provided as validation of the proposed system.

Keywords: file dispersal, P2P networks, DHT, mobile systems, erasure coding

1 Introduction

In many contexts a lightweight approach to information management is more suitable than reliance on a database management system. Unstructured data with infrequent search operations such as reports and confidential documents present an important area where lightweight file management is appropriate. This particular perspective on file operations has seen the development of various schemes for network and distributed storage [1,2]. The availability of P2P systems and distributed schemes for file storage has given a greater impetus to this mode of management. The ability of these models to overcome resource heterogeneity is part of their appeal and relevance. The increased flexibility and enhanced functionality that result from the convergence of P2P models and mobile devices is however hampered by concerns over security, reliability and efficiency. In P2P systems specific security and authentication mechanisms [3] are called upon to mitigate the vulnerability that result from the symmetric role they play and their *ad hoc* configuration and evolution. These obligations are further compounded by the restrictions that limited battery life in mobile devices imposes on processing, storage and bandwidth [4]. Negotiating a design that ensures resilience has become a core requirement for a network file management in a mobile P2P network. Data redundancy is often used as an effective method for ensuring reliability, and is manifest in two main techniques: pure replication and erasure coding. Replication can take two forms: monolithic or fragmented. In monolithic replication the replicas of an entire file, as monolithic blocs, are distributed to different nodes of a network [5]. The file may be encrypted before distribution and the retrieval of a file can be from any node. This type of replication may be vulnerable to intensive cryptanalysis because of the availability of the entire file in one single node. In fragmented replication, on the other hand, a file is split into fragments, which are then dispatched to different nodes. This is performed by an information dispersal algorithm (IDA) [6]. Although this scheme is relatively secure it can be vulnerable to the withdrawal or failure of a participating node since the reconstruction of the entire file requires all the fragments to be available. A high level of replication in files and file fragments can, to some extent, alleviate this problem.

Erasure coding was introduced as an optimisation of IDA schemes; only a subset of the dispersed fragments is needed in order to reconstruct the original file [7]. Under this scheme a file is split into s initial fragments (segments). The encoding scheme generates $s+r$ fragments such that the original file can be reconstructed from any s of the $s+r$ encoded fragments. In a P2P network each fragment is allocated to one peer. Erasure coding is considered particularly useful in unstable environments and especially in P2P systems. It was demonstrated that the mean time to failure (MTTF) is much higher that for replication with the same storage overhead and repair period [8]. It was also noted, however, that the inherent complexity of erasure coding might outweigh its advantages [9]. In contrast, replication

Figure 1. System architecture

in its monolithic or fragmented form may be more efficient, easier to implement and more suitable in stable environments.

There is little work reported on the implementation of file dispersal on mobile devices. The focus of most of the research in this area has been on the implementation of file sharing environments and the development of structured DHT-based P2P mobile systems [10, 11]. The main contribution of this work is in the development of an experimental mobile P2P framework that addresses and reconciles P2P and mobiles issues in file dispersal. In particular, this paper is concerned with architecture for supporting the two methods for file dispersal on a mobile P2P system, pure replication and erasure coding

The remainder of the paper is structured as follows. Section 2 presents the main components of the system architecture. Section 3 provides some validation of the system through experimental results. Section 4 raises some issues for discussion and Section 5 concludes the paper.

2 System architecture

The mobile P2P system should support three main operations on a file: file dispersal, file retrieval and file removal. As file dispersal is the most important function this paper will focus on its architectural features and on the steps involved in its deployment. All the steps, except fragment encoding, are common to both pure replication and erasure coding:

- File selection
- File compression
- File fragmentation
- Fragment encoding (for erasure coding only)
- Fragment encryption
- Fragment distribution
- Metadata encryption and its secure storage

The role of the system is to support these operations securely and efficiently.

2.1 System Components

The clear separation of concerns that the system postulates is better served by a layered architecture (Figure 1). Each layer is enhanced by service procedures. At the highest level is the file dispersal application, which relies on a structured DHT-based P2P network. The lowest layer deals with communications, which are handled by TCP. Security issues are attended to at all levels. CP-based communications may incur some overheads but the error free channels that are generated contribute to the reliability of the file dispersal process.

File fragmentation and distribution were key factors

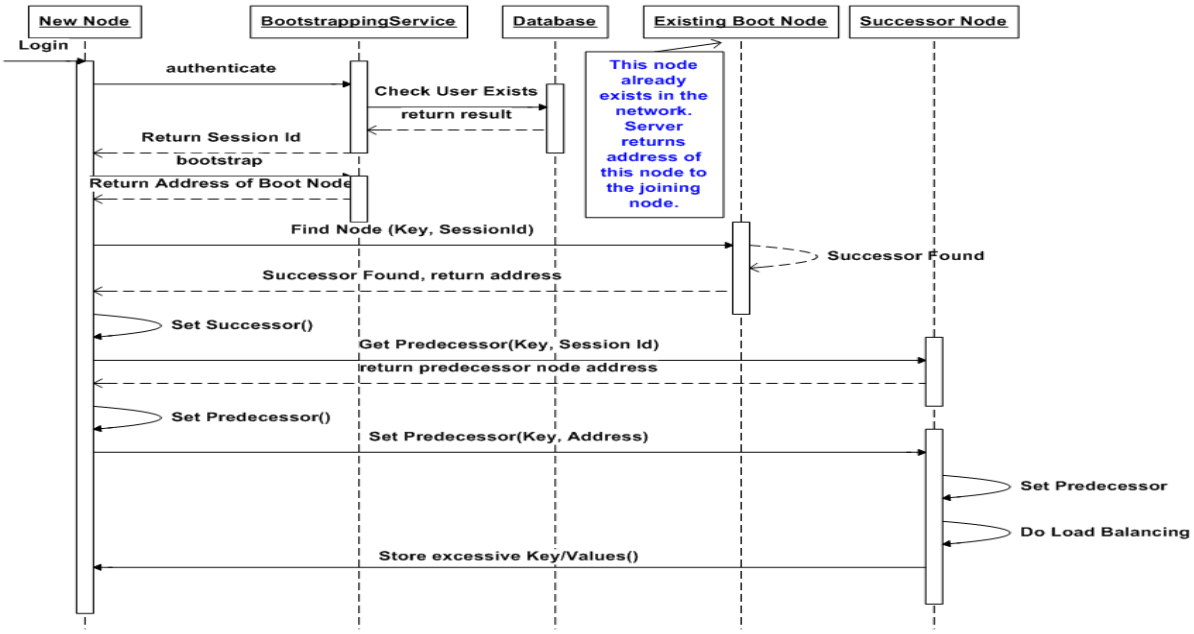

Figure 2. Bootstrap process

Figure 3. Chord finger table

in the choice of a structured DHT-based approach to P2P implementation. As an implementation of a P2P network, distributed hash tables (DHT) offer intuitive mapping from file fragments to nodes. The ID of a node can be generated by applying a hash function such as SHA2 to its IP address. Similarly a fragment can be associated with an ID created from its content by the hash function. Nodes are allocated an ID space for the fragments whose hash values fall within that space. The advantages of overlay networks include their ability to deal with heterogeneous systems and their capacity for self-management and reorganisation. The Chord overlay network was chosen because of its simplicity, its scalability and its ability to guarantee efficient convergence in search queries, which is *O(logN)*, where *N* is the number of nodes in the network [12]. Its relative efficiency is also a valuable attribute in a mobile context. A Chord overlay network is characterised by a ring topology where every peer has a predecessor and a successor. Peers are arranged in a circle and addresses range from *0* to $2^m - 1$. Peers keep track of other peers in the network through a routing table, the finger table, whose size is equal to at most *m*. Kademlia is another DHT-based overlay network with a tree topology. One of its advantages is the way it deals with the arrival of new nodes. Older nodes are always given priority.

In the proposed system Chord was enhanced with the Kademlia's method for managing the finger table. The dynamics of a node entry into the Chord-based network are presented in the sequence diagram of the bootstrap process (Figure 2). It depicts the way Chord accommodates a new node in the routing table by identifying and setting appropriately the successor and predecessor nodes. The user is able to view details of the finger table of Chord (Figure 3).

2. 2 Implementation

At P2P (DHT) level authentication and secure communication are the main concerns. For authentication a session-based procedure was chosen, thanks to the ease with which it can be incorporated into the bootstrap service. A central database server is used to help the bootstrap service authenticate incoming users. Secure communication is enforced by a number of measures. Both symmetric and asymmetric encryption schemes were implemented. For fragment encryption six symmetric algorithms can be applied randomly to the fragments, in order to further hamper any cryptanalysis. The integrity of the messages between peers is assured by MD5 message digests. Moreover, once a file is successfully dispersed, its metadata is encrypted and stored securely on the local host.

In the design, efficiency considerations were driven by processing, memory and battery constraints. For example, lightweight versions of encryption algorithms and symmetric schemes were selected where appropriate. The erasure coding function is an adaptation in J2ME of the Java version used in JigDFS [7]. The system was implemented on the J2ME platform and deployed on Sun Java Wireless toolkit 2.5.2 for the Connected Limited Device Configuration (CLDC).

3 Experimental results

A number of experiments were conducted in order to shed more light on system behaviour at different stages of the file dispersal process. These refer specifically to the key generation from fragments, the performance of the encryption algorithms and the resilience of the system.

3.1 Hash key distribution

The hashing function has to be consistent and the hash key space generation should be equally distributed so that the keys do not map onto only one subset of the peers. Moreover the hash function should generate hash keys, which are collision resistant so that there is a low probability that two peers in the network are associated with the same key or two different file fragments generate the same key. The key distribution was investigated by applying the fragmentation and hash generation to a file. A random text file with 848 words was used as an input to generate hash keys and the output was used to plot the graph in Figure 4. Overall the hashing results seem to confirm the viability of the system.

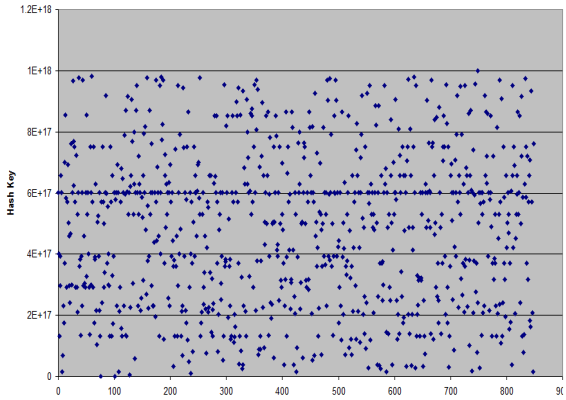

Figure 4. Hash key distribution

3.2 Encryption/Decryption

The relative performance of the encryption algorithms was analysed in order to gain an insight into their impact on the processing overheads on mobile devices. In the encryption/decryption time split graph, the percentage of time taken varies with the algorithms (Figure 5). The actual average values, in milliseconds taken for the execution run-time of the algorithms during the experiment are shown in their respective columns. Almost all of the algorithms fall within 40%-60% range, which indicates that their encryption and decryption processing times are comparable and that both the receiving and the sending peers incur the same processing penalties. DES is the only algorithm, which takes more time during encryption than during decryption. It takes altogether much less time than the other algorithms and it can be assumed that it does not affect the overall behaviour of the system.

3.3 Resilience Analysis

A higher level of replication increases the robustness of the system so that all the fragments of the file can still be recovered despite the absence of some peers. To evaluate the effectiveness of replication, an experiment was performed with 5 peers. One peer is used for dispersing and downloading the file throughout the experiment so that the results are consistent. A file of size 14846 bytes was used with the default fragmentation settings.

In Figure 6 the graph displays the experimental results for the same file with different replication levels and a different number of offline peers. For offline peers, the peer selection was random so that a non-deterministic behaviour of the network peers can be simulated. In the presentation of the results in the graph, 1 is used when a file is completely downloaded, 0 otherwise. A value on the graph indicates that the file was successfully reconstructed. The graph indicates that the highest number of file retrievals occurred when the highest replication level was set. In the case of erasure coding, although four peers were offline it appears that there were enough replicated fragments that resided on the store of the owner. With no replication, even when one single peer was offline, the file could not be recovered in full; the peer, which was offline, had some of the fragments that were needed by the owner to rebuild the entire file. This experiment confirms also that with a threshold of 3, erasure coding is always successful in reconstructing the original file. The relevance of the provision of the two modes of data redundancy is supported by the experimental results.

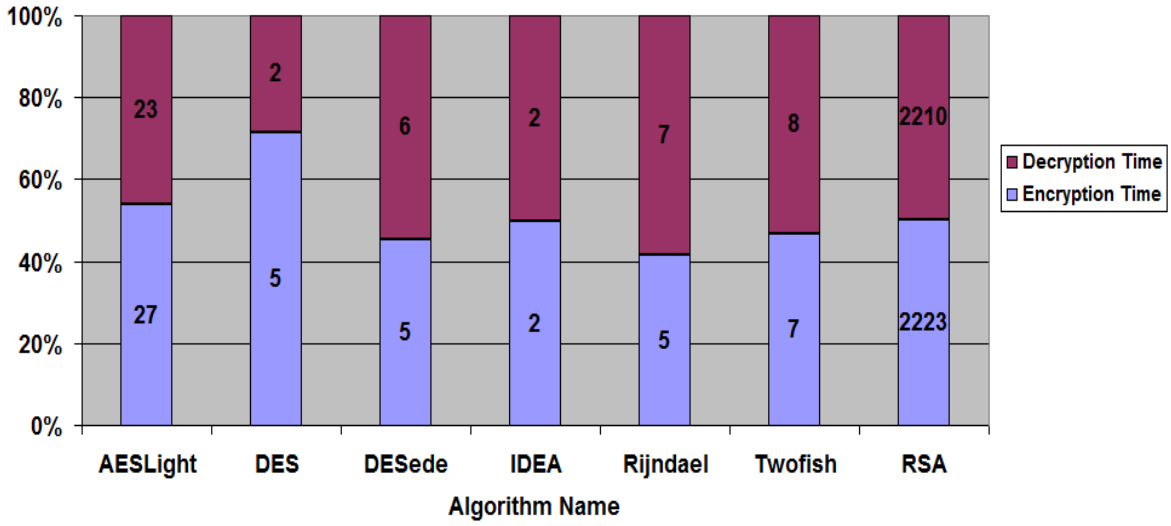

Figure 5. Encryption and decryption

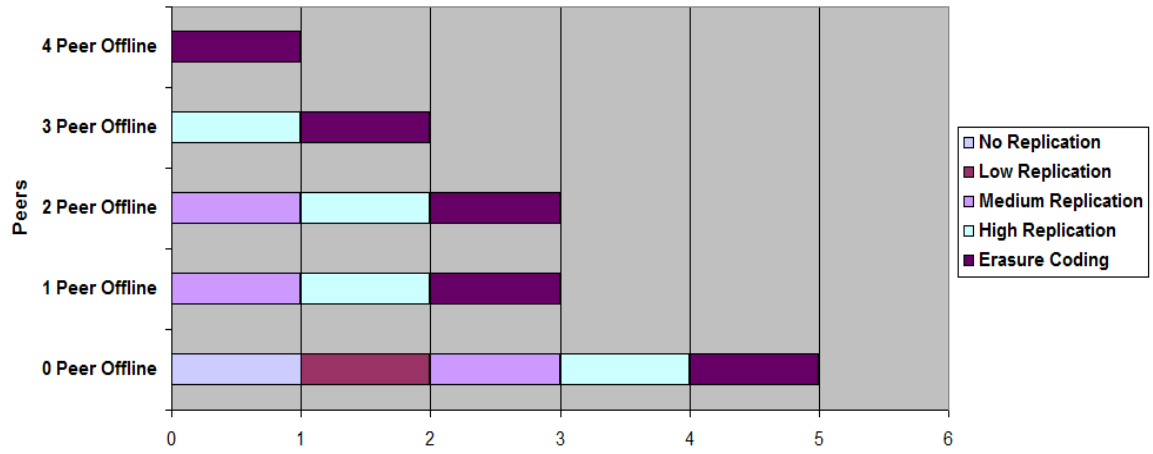

Figure 6. System resilience

4 Discussion

Most of the work on file dispersal has been on wired networks, where many existing systems deal with varying degrees of file granularity and encryption [7, 13, 14]. A mobile P2P context for file dispersal is relatively novel but it presents a number of challenges that the proposed system has addressed through the perspectives of efficiency and security. These issues have shaped its architectural features and determined the selection and adaptation of algorithms.

4.1 Mobility

Mobility requirements and constraints in a P2P environment were considered at two levels. At the application level the efficiency concerns was addressed in the approach to the design of the system. Two forms of replication were included, pure replication and erasure coding, in addition to optimisation options such as compression. This issue was also a guiding factor in the selection and implementation of light algorithms and their performance.

4.2 Flexibility

Flexibility and configuration represent another facet of the system. One way of dealing with the openness of P2P systems, their dynamic and ad hoc nature and their potential instability is by designing a flexible system and by providing a suitable interface for configuration (Figure 7). Confidentiality is already addressed by the fragmentation itself and by the random application of encryption algorithms. The system offers flexibility by providing users with two methods for performing file dispersal to suit environmental conditions. Another aspect of flexibility concerns the portability that follows from the choice of J2ME as a platform for implementation. This covers ease of redeployment and resilience to heterogeneity.

4.3 Network architecture

The network architecture and configuration has undoubtedly played a critical role in the fulfilment of system requirements. The management of a DHT-based overlay network has additional benefits for reliability and efficiency. In particular,

1. the resilience of the system through CHORD capacity for self-management and the minimisation of DOS attacks through the adequate management of the routing tables.
2. the handling of the heterogeneity of nodes

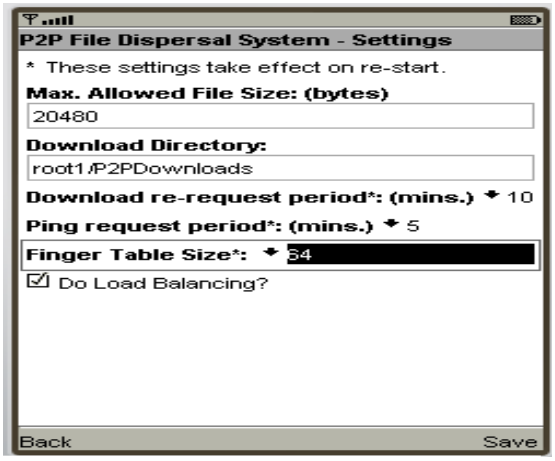

Figure 7. System configuration

through widely deployed and supported system software.

3. the direct mapping of fragments to nodes and the efficiency of the node search.

4.4 Further work

One notable contribution on the deployment of DHT-based mobile P2P systems and their performance is reported in [16], where the authors present the performance of a working prototype. The research indicates that the processing of incoming messages requires a lot of processing, and offers pointers for improvement to the proposed system. The statistical analysis has shed some light on the performance of the system, and has underlined the importance of the optimisation of the functional components that contribute to the file dispersal process. Specific functions that require attention include bootstrapping and downloading. Further work will also involve the integration of a trust-based system and the refinement of the authentication process. The system will also benefit from a finer configuration in the selection and use of the encryption algorithms.

5 Conclusion

An approach to reliable file dispersal in mobile P2P networks was presented in this paper through the implementation of two methods, pure replication and erasure coding. The reliability of the system was assured by a layered architecture and by various mechanisms that address concerns over security in P2P systems and efficiency in mobile devices. The provision of alternative schemes for file dispersal, the inclusion of lightweight algorithms and the grafting of an authentication service enhanced by secure communications all contribute to the creation of a viable file dispersal system.

6 References

[1] Ye W., Khan A.I. and Kendall E.A. Distributed network file storage for a serverless (P2P) network. *The 11th IEEE International Conference on Networks (ICON2003)*, Sydney, Australia, 2003, pp343-347

[2] Sheng, B, Li, Q and Mao, W. Optimize Storage Placement in Sensor Networks, *IEEE Transactions on Mobile Computing*, Issue: 99, 2010, pp1-14

[3] Androutsellis-Theotokis, S., and Spinellis, D. A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys , 36 (4),* December 2004, pp335-371.

[4] Nurminen J.K. and Nöyränen J., Energy-Consumption in Mobile Peer-to-Peer - Quantitative Results from File Sharing, *5th IEEE Consumer Communications & Networking Conference (CCNC)*, Las Vegas, USA, January 2008, pp730-733.

[5] Renuga, K., Tan, S.S., Zhu, Y.Q., Low, T.C., Wang, Y.H. Balanced and Efficient Data Placement and Replication Strategy for Distributed Backup Storage Systems. *International Conference on Computational Science and Engineering (CSE '09)*, Volume: 1, 2009, pp87-94

[6] Dabek F., Kaashoek M.F., Karger D., Morris R. and Stoica I., Wide-area cooperative storage with CFS, *Proceedings of 18th ACM symposium on operating systems principles SOSP* (2001).

[7] Bian J. and Seker R. "JigDFS: A Secure Distributed File System", *IEEE Symposium* on *Computational Intelligence in Cyber Security CICS '09.* March/April 2009, pp.76-82.

[8] *IEEE Symposium on Computational Intelligence in Cyber Security* (CICS '09), April 2 2009, Nashville, USA, pp76–82.

[9] Weatherspoon H., Kubiatowicz J., Erasure Coding Vs. Replication: A Quantitative Comparison, Revised Papers from the First International *Workshop on Peer-to-Peer Systems*, March 07-08, 2002, p328-338.

[10] Rodrigues R. and Liskov B.. High Availability in DHTs: Erasure Coding vs. Replication, *Peer-to-Peer Systems IV 4th International Workshop (IPTPS 2005*), Ithaca, New York, February 2005.

[11] McNamara, G. and Yanyan Yang, Creating a mobile file sharing environment over Bluetooth, *Third International Conference on Pervasive Computing and Applications, (CPCA 2008),* Volume 2, 2008, pp 863-868

[12] Galluccio, L.; Palazzo, S.; Rametta, C., On the efficiency and trustworthiness of DHT-based P2P search algorithms in mobile wireless networks. *International Conference on Ultra Modern Telecommunications & Workshops, (ICUMT '09),* 2009, pp1-8.

[13] Stoica I., Morris R., Karger D., Kaashoek M.F., Balakrishnan H., "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", MIT Laboratory for Computer Science, *SIGCOMM'01*, August 2001, San Diego, California, USA. *http://pdos.lcs.mit.edu/chord/.*

[14] The OceanStore Project. *http://oceanstore.cs.berkeley.edu/*

[15] Amann B., Elser B., Houri Y. and Fuhrmann. IgorFs: A Distributed P2P File System. *2008 Eighth International Conference on Peer-to-Peer Computing*, Aachen, Germany, 2008, pp.77-78.

[16] Kelényi I, Nurminen J.K: Energy Aspects of Peer Cooperation - Measurements with a Mobile DHT System. *43th IEEE International Conference on Communications (ICC 2008)*, Beijing, 2008.