P2Pを利用したデータバックアップシステムの提案

吉田哲平 *　　小高知宏 *　黒岩丈介 **　　白井治彦 ***

Fault Tolerant Data Backup System Using Peer-to-Peer Network

Teppei YOSHIDA*, Tomohiro ODAKA*,
Jousuke KUROIWA and Haruhiko SHIRAI*****

(Received February 5, 2016)

In this study, we investigate the backup-system using peer-to-peer network system. In general, as a way to backup the data, we use magnetic tape, hard-disk, flash memory, file server, and online storage, etc. However, these devices have some problems on redundancy, dispersiveness, or security.

We propose fault tolerant data backup system with peer-to-peer network. To improve the level of redundancy and dispersiveness, our system employs the number of peer-to-peer network's nodes and redundant encrypting backup data. We attempt to reduce the nodes' load by halving backup data and sending it to separate nodes.

Key words **:** Peer-to-Peer, Data Backup, Fault Tolerance, Redundancy, Dispersiveness, Security

1. はじめに

本研究では，個人や企業などで用いられるデータのバックアップを行う時，災害や障害に対して強固なデータバックアップシステムを構築することを目的とする．

従来行われているバックアップの手段として，磁気テープ，ハードディスク，フラッシュメモリ，ファイルサーバ，オンラインストレージなどを利用したデータの管理が存在する．磁気テープやハードディスクはコストが低い一方，物理的な力に弱い[1][2]．フラッシュメモリは比較的高速であるが，書き込むことにより劣化する．ファイルサーバやオンラインストレージはデータの一括管理が可能であるが，障害やセキュリティ面に不安を残す．

*大学院工学研究科原子力・エネルギー安全工学専攻
**大学院工学研究科知能システム工学専攻
*** 工学部技術部
*Nuclear Power and Energy Safety Engineering Course, Graduate School of Engineering
**Human and Artificial Intelligent Systems Course, Graduate School of Engineering
***Technical Division

これらのバックアップの手段で共通して問題となってくることは，冗長性，分散性，セキュリティの3つである．冗長性は，障害に備えて余剰の予備装置を用意しておく度合いである．冗長性の問題は記憶媒体を増やすことで解決することができる．しかし，記憶媒体を増加させることはコストが嵩み，分散性の問題を解決することができない．本研究での分散性は，データが複数存在する場合，データの記憶媒体が地理的に分散している度合いを指す．磁気テープなどの一個人が管理する形態では，分散性を高めるには移動方法，管理場所などを一個人が把握していないといけないので非常に手間である．セキュリティに関しては，記憶媒体の管理方法が杜撰であったり，オンラインストレージにおいては利用する際のIDやパスワードが紛失してしまったりする恐れがある．

そこで本研究では，Peer to Peerネットワークを利用した障害に強いデータバックアップシステムを提案する．このPeer to Peer通信を用いると障害に強いシステムを構築することできる．この通信方式を利用することで，従来のバックアップ手法で問題提起した冗長性，分散性，セキュリティの3つを解決できるのではないかと考えられる．冗長性と分散性はPeer to Peerネットワーク上に存在するノードの数を増やすことによって

向上する．また，セキュリティはデータを暗号化することによって向上させる．

しかし，ネットワーク上の各々のノードにそのままのデータで保存しておくのは一つのノードのストレージに対して，負荷が掛かってしまう可能性がある．本研究では，一つのデータを二分割し，それぞれのデータをネットワーク上の別のノードに保存することにより，一つのノードの負担を軽減を行った．しかしながら，一つのデータを保存しているノードに障害が起こった場合，二分割する前の元のデータに復元することができなくなるという問題が発生する．そこで，本研究では排他的論理和を二分割したデータのバイナリに用いることで解決する．

本稿は，2 章で従来のバックアップ手法と問題点，本システムの概要について述べる．3 章では，本システムの設計について述べる．4 章では，考察について述べ，5 章で本研究のまとめを述べる．

2. 従来のバックアップ手法と問題点

2.1 従来のバックアップ手法

従来使われているバックアップ手法として以下のようなものが存在する．

磁気テープ 音声や映像信号などのアナログ信号や計算機のデジタル信号などの波形の強弱を磁化し，記録して再生することのできるテープ上の磁気記録媒体である．この媒体の特徴として，安価であり，保存時に電源を必要としないことから長期保存に向いている．一方で，磁気や物理的な力に弱く，定期的なメンテナンスを必要とする．また，ランダムアクセスができないということから，一部のデータだけリストアする用途には向いていない．

ハードディスク 内部の磁気ディスクをモーターで高速に回転させ，磁気ヘッドに近づけることでデータの読み書きする記憶装置である．磁気テープと同じような特徴を持っているが，こちらはランダムアクセスが可能で，振動に弱い．

光ディスク レーザー光によってデータの読み書きを行う記憶媒体である．CD，DVD，Blu-ray Disc などの種類があり，容量があまり大きくないが，寿命が長く取り扱いが容易，また製造コストを抑えられるため音楽や映像などを記録した記憶媒体として普及している．

フラッシュメモリ フラッシュメモリのセル (記憶素子) の浮遊ゲートに電子を封入することでデータを記憶する媒体である．読み書きが比較的高速であるがこの電子によってセルが劣化するため，書き込み可能回数が限られてくる．USB メモリや SSD などの種類があり，近年では書き込み可能回数が改善されてきている．

ファイルサーバ ネットワークを利用して外部のコンピュータとデータの読み書きが行えるコンピュータである．個人や組織のデータの一括管理が可能であるが，導入を一から行うと困難で利用する人数が多くなるほど誤操作などによって保存データを消失する可能が高くなる．また，ネットワークに障害が起こると使用することができなくなる．

オンラインストレージ 特定の組織が個人にディスクスペースを貸し出し，ファイルをアップロードすることでバックアップを行えるオンラインサービスである．他人とのファイル共有も可能であるが，データの流出やサービス停止によるデータの消失などが懸念される．また，悪意のあるユーザがオンラインストレージのあるユーザのログイン ID やパスワードを入手してしまうと，そこに保存されているファイルを閲覧・書き換えができるので，これを狙ったフィッシングサイトも横行している．

2.2 従来のバックアップ手法における問題点

前節で挙げられているバックアップ手法は一般的に取られているものである．しかし，これらには冗長性，分散性，セキュリティが不十分ではないかと考えられる．そこで，この 3 つを向上させる手法を考案する．

まず冗長性に関しては，一つの記憶媒体でデータを管理していた場合，その装置がいつ故障してしまうかを考えなければならない．このとき，データを管理する記憶媒体を増やすことで，一つの装置が故障したとしても復旧できる可能性を増やすことが出来る．しかし，装置を増やすことでコストが増加する．

また，分散性に関しては，特定の場所だけで記憶媒体を管理する場合，地震や火事の災害などによってデータごと消失する可能性がある．この解決方法として，物理的に記憶媒体を移動する方法が考えられるが，非常に手間で移動するコストもかかってしまう．また，分散性を更に高めようとすると，一個人では管理する場所が賄えなくなってしまう．

最後にセキュリティに関しては，光ディスクやフラッシュメモリの場合，手軽に持ち出せるため，紛失する可能性が高くなる．したがって，何らかの対策を講じていないと他人にファイルの中身を閲覧されてしまう．また，オンラインストレージなどでログイン ID やパ

スワードが流出してしまった場合は，ネットワーク上の人間に悪用されてしまう可能性があり非常に危険である．

2.3 Peer to Peer ネットワークを利用したバックアップシステム

本研究では，Peer to Peer ネットワークを利用したバックアップ手法を提案する．Peer to Peer とは，ネットワークを形成するピア，またはノードと呼ばれる端末が対等の立場で通信しあう方式である．この Peer to Peer 通信の端末を更に増やしていくことによって，作られてくるものが Peer to Peer ネットワークである．

この通信方式を利用したものとして，Gunutella[3] ,Napster,WinMX,Winny[4] ,Skype[5] などがある．しかし，著作権上などの問題から Napster や WinMX はサービスが停止 [6] され，WinMX に至っては逮捕者が出ている [7] .

本研究では，特定のネットワーク上に，個人が信頼できる人で構成されたグループ内での使用を想定した Peer to Peer ネットワークを形成する．このネットワークを形成することによって，ファイルをネットワーク上の端末ごとにアップロードするシステムを構成する．このシステムにより，前節で挙げられた問題点が解決できるのではないかと考えられる．

冗長性に関しては，Peer to Peer ネットワーク上に参加している端末の数だけ冗長性を高くできるので，信頼性が増加する．また，参加しているユーザは複数台持つことなくネットワーク上の参加している端末にデータを置くことができるので，個人のコスト削減にも繋がる．

また，分散性に関しては，Peer to Peer ネットワークに参加しているユーザが，それぞれ地理的に離れていれば分散性が増す．このとき，ファイルの移動方法はネットワークを介した通信であるので非常に簡単である．そして，災害などによって一つの端末が消失したとしても，他の端末にデータが残っていれば復旧することができる．

セキュリティに関しては，個人が信頼できる人で構成されたグループ内で使用を想定しているが，他人に見られてはいけないファイルも存在するため暗号化処理を行う必要がある．この暗号化処理はファイルを圧縮する際に行う．また，保存したノードの所有者にデータを改竄される恐れがあり，これを検出する手法として誤り訂正符号の BCH を用いたシステム [8][9] が存在するが大きなデータに対して処理時間が掛かってしまうため，今回はハッシュを使用した手法を用いた．

また，本研究では一つのファイルを二分割し，その二つのバイナリファイルの排他的論理和を計算することでパリティファイルというものを作成する．このパリティファイルを作成することにより，二分割した片方のバイナリファイルが Peer to Peer ネットワーク上から消失したとしても，残っている片方のバイナリファイルとパリティファイルの排他的論理和を計算することにより，消失した片方のバイナリファイルを復元することができる．この 3 つのファイルを Peer to Peer ネットワーク上に分散させることによって，ネットワーク上の端末が同時に 2 つ以上壊れない限りは復元することができるため，障害に強いシステムが構築できるのではないかと考えられる．

3. 設計方法

本章では本システムの設計方法について述べる．本システムは Peer to Peer を用いたデータバックアップシステムを実装する．本システムで実装する機能は，バックアップ部，ネットワーク部，リストア部，ユーザインターフェース部，内部処理部の 5 つの機能に分かれている．バックアップ部はファイルの暗号化やファイルの分割，パリティを生成することでバックアップを可能とする．ネットワーク部はファイルの送信や検索を行うことでネットワーク上の処理を行う．リストア部はバックアップ部で処理を行ったデータを元のバックアップデータに戻すことでリストアすることを可能にする．ユーザインターフェース部はユーザがファイルの検索や送信の命令をシステムに出すことのできるインターフェースを実装する．内部処理部はスレッドを使うことで並列処理を行う．処理の内容や手順については以下で述べる．

また，バックアップ処理からリストア処理までの一連の処理の流れを図 1 に示す．

1. まず，本システムのバックアップ部が本システムの同じ階層の backup ディレクトリを監視しているため，バックアップファイルを backup ディレクトリに置くことでバックアップ部の処理を開始する．

2. バックアップ部ではファイルの暗号化やファイルの分割，パリティの生成処理などを行ったあと，3 つのファイルは cache ディレクトリに置かれる．

3. その後，ユーザがシステムのユーザインタフェースで操作することによりネットワーク部でのファイルの送信が行われる．

4. バックアップファイルを自分の端末に戻す時は，ユーザがユーザインタフェースからネットワーク部

へファイルの検索の命令をすることで目的のファイルを探し出し，探しだすことができれば取得する．

5. 取得後，リストア部によってファイル結合などをすることで元のファイルに戻すことができる．

また，本システムでは KEY と UID という固有の ID を用いることでファイルの暗号化やファイルの検証などを行う．

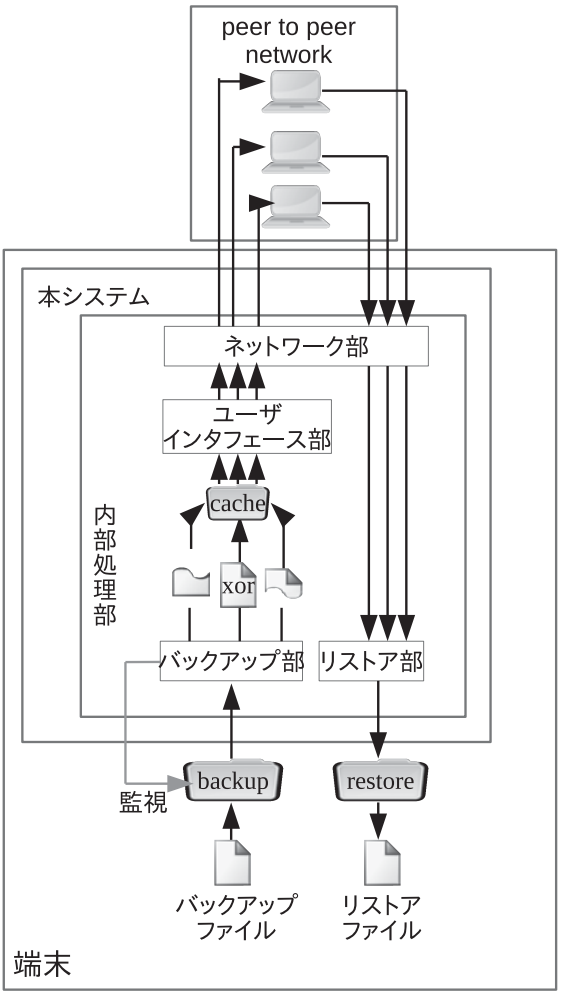

図 1　システムの全体図とバックアップからリストアまでの流れ

3.1　バックアップ部

バックアップ部では，ネットワーク上にファイルをバックアップするために必要な処理を行う．バックアップ部のフローチャートを図 2 に示す．以下に示される工程を本研究ではキャッシュ化と呼び，キャッシュ化されたファイルをキャッシュファイルと呼ぶ．プログラムが開始されると，backup ディレクトリの監視をする．このとき，backup ディレクトリにファイルが置かれると，そのファイルを二分割し，二分割したファイルのそれぞれを backup ディレクトリを図 2 の最下段に当たる cache ディレクトリにキャッシュファイルを保存することによりキャッシュ化が完了する．

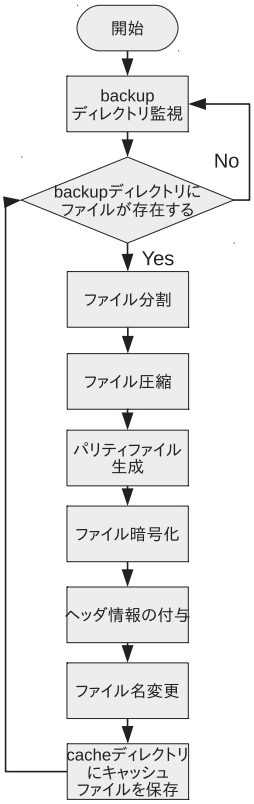

図 2　キャッシュ化のフローチャート

図 2 で示される工程を以下で述べる．

ファイルの分割　バックアップを行うファイルをそのままで Peer to Peer ネットワーク上の他の端末に送信した場合，送信先の端末が故障するとファイルの復旧を行えなくなる．この解決法として，送信先の端末を増やすことで冗長性を高めることが挙げられるが，その分送信先の各端末の負担が増えてしまう．これを軽減するために，ファイルを二分割しファイル容量を減らすことによって解決する．

ファイルの圧縮　バックアップを行うファイルは容量の小さいものから大きいものまで存在する．そのため，そのままのファイルでは，通信量が多くなったり保存先の端末のストレージに負担がかかることからファイルの圧縮を行う．本研究では ZIP 形式で圧縮を行う．

パリティファイルの生成　ファイルを二分割して二つのファイルを送信する場合，送信先の一つの端末が故障するとファイルの復旧ができなくなる．そのため，新たにもう一つのパリティファイルという

ものを作成する必要がある．このパリティファイルは二分割したファイルの二つのバイナリの排他的論理和を計算し，作成したファイルである．このパリティファイルを作成することにより，二分割したファイルの片方を管理している端末が故障したとしても，二分割したファイルのもう片方とパリティファイルの排他的論理和を計算することによって，ファイルの片方の復旧が可能となる．

ファイルの暗号化 生のデータでインターネット上にファイルを保存した場合，インターネットに接続した者は誰でもファイルの中身を閲覧することが出来る．悪意あるユーザからファイルの中身を閲覧できないように AES-256[10] によって暗号化を施す．

ヘッダ情報の付与 ファイルをそのまま Peer to Peer ネットワーク上の他の端末に送信した場合，どのファイルが誰のものかわからなくなってしまう恐れがある．そのため，暗号化されたファイルのヘッダ上に所有者の ID やハッシュ値を書き込む処理を行う．

ファイル名の変更 一般的にユーザがつけるファイル名は多彩であるが，ファイルを送信する際に，Peer to Peer ネットワーク上の他の端末に保存されているファイルと送信するファイルの名前が同じになる恐れがある．この事態を防ぐために，ファイルを送信する前にシステムが指定した規則でファイル名を決定する．

3.2 リストア部

リストア部では，バックアップファイルをネットワーク上に送信した後，自分が使用している端末に戻すときの処理を行う．リストア部はネットワークから自分のキャッシュファイルを取得した後に行う処理である．リストア部の処理の流れを図 3 に示す．この処理は二分割されたキャッシュファイルを同時に処理する．処理開始後，改行コードを利用してキャッシュファイルのヘッダ情報とデータに分けた後，ヘッダ情報から UID を取得し，自分の UID と異なっていた場合，この後の処理を行わないこととする．次に KEY を使用してファイルを復号する．パスワードが異なった場合はこの後の処理は行わないこととする．ファイルの複合に成功するとファイルの解凍を行う．ファイルを解凍後，ファイルが自分のバックアップしたファイルであるかどうかファイルの検証を行い，最後に restore ディレクトリに元のファイルが保存される．

ヘッダ情報解析 ファイルをキャッシュ化する際に，ファイルにヘッダ情報を書き込んだため，この情報を手がかりに元のファイル名や誰が所有しているファイルであるかを確認することが出来る．

ファイル復号 キャッシュ化したファイルは暗号化されているため，そのままの状態では扱うことが出来ない．そのため，復号する必要がある．キャッシュファイルに入力したパスワードと誤っていた場合，以後の処理は行わない．

ファイル解凍 キャッシュ化したファイルは圧縮されているため，そのままの状態では扱うことが出来ない．そのため，解凍する必要がある．本研究では ZIP 形式のファイルの解凍を行う．

ファイル検証 バックアップを行うファイルは Peer to Peer ネットワーク上の自分が所有している端末とは違う別の端末へと保存される．そのため，信頼できる Peer to Peer ネットワークとはいえ，ファイルの送信先の端末の所有者がファイルを改竄する可能性がある．これを防ぐために，ヘッダ情報を付与する際にハッシュ値を計算し，ファイルのヘッダ上に書き込んだ．これを行うことによって，リストアしたファイルとファイルのヘッダ上のハッシュ値を比較することでファイルが改竄されているか検知することができる．

ファイル結合 二分割ファイルはそのままで扱うことはできないため，二分割したファイルを結合する必要がある．二分割したファイルのバイナリを結合することで復元する．

3.3 ネットワーク部

本システムでは，Peer to Peer を利用したネットワークを実装する．その上で以下の機能を実装する必要がある．Peer to Peer ネットワークには様々な種類のプロトコルが存在するが，本研究では Chord[11] というプロトコルを用いてネットワークを実装する．Chord は図 4 で示すような環状のネットワークをイメージしたハッシュ空間を探索することで，探索効率や確実性が高い．図 4 では，それぞれのノードに端末 ID が振られルーティングがなされている．

このプロトコルを実装するために以下の機能が必要となる．

ネットワークへの参加 ユーザが本システムを利用してバックアップを行う際に，端末を本システムのネットワークに参加させる必要がある．ネットワークに参加するために，参加したいネットワーク上の

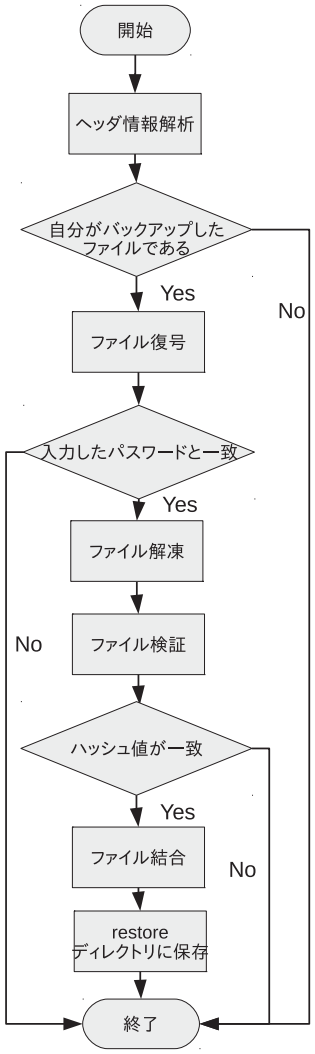

図 3　リストアのフローチャート

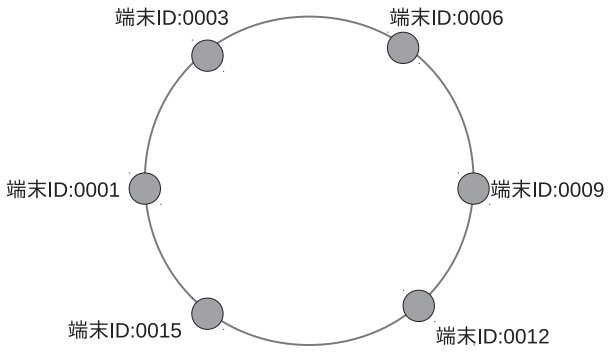

図 4　Chord のイメージ図

一つのノードの情報を知る必要がある．このノードのことを本研究では初期ノードと呼ぶ．

ネットワークからの離脱　ネットワーク上に参加している端末を本システムを終了させるときなどにネットワークから離脱させる必要がある．また，システム上にネットワークから離脱する機能を作ったとしても，端末の電源を切ることでネットワークを物理的に遮断する場合もあるのでそれを考慮した設計が必要となる．

ルーティングの更新　ネットワークの各々のノードはネットワーク上のどのノードと繋がれているかを知っておかなければならない．そのため，目的のノードに達するための最適なルーティング情報を持つ必要がある．また，ネットワーク上ではノードの参加･離脱が頻繁に行われる可能性があるため，ルーティング情報を定期的に更新する必要がある．

ファイル送信　データのバックアップはネットワーク上のノードにファイルを送信することで完了する．本研究では，二分割したファイルとパリティファイルを分けなければならないので，ネットワーク上に自分が使用しているノード意外に最低 3 つ存在しなければならない．

ファイル検索　ファイルをリストアするときに，ネットワーク上のノードからファイルを取得する必要がある．そのため，ネットワーク上のどのノードが自分のファイルを持っているかを検索できる機能が必要である．このとき，二つのキャッシュファイルとパリティファイルの 3 つのファイルが本来あるべきノードにないとき，ファイルを復元する機能が必要となる．

3.4　ユーザインターフェース部

本システムを使用するユーザがバックアップ機能やリストア機能などを使用するときに必要なインターフェースを実装する必要がある．

また，ネットワークに参加する際の初期ノードを登録するためのポート番号や IP アドレスを入力する欄が必要である．

更に現在どのノードと接続されているかの確認，キャッシュファイルの表示，リストアの状況表示，ログの表示を出力させなければならない．

4. 考察

本システムは，Peer to Peer を利用することで手軽にバックアップを行うことができると考えられる．その一方，手軽さ故にデータの送信先が悪意のあるユーザでデータを悪用される危険性も孕んでいる．しかし本研究では，信頼されたユーザでネットワークを構築することを想定しており，その危険性は低いと考えられる．

また，ファイルを二分割し，パリティファイルを生成することによって冗長性・分散性が増し，ファイルが消失したとしても復元できる可能性が高くなる．ただし，その分ネットワーク上のノード数を増やさなければ信頼性が失われるため，少人数で運用していく場合，ノード一つあたりの責任の比重が重くなってしまう．

システムを運用する上で，考えられる問題点として，他のユーザのストレージの容量を考慮していないため，送信先の容量が少なくなっている場合，ストレージを圧迫してしまう可能性がある．そのため，ユーザは予めバックアップ領域を決めておき，キャッシュファイルをネットワーク上に送信する時，システムが動的にファイルの送信先を決定するべきだと考えられる．

5. まとめ

本研究は Peer to Peer ネットワークを利用して障害に強いデータバックアップシステムの構築を行った．従来のバックアップ手法である，磁気テープ，ハードディスク，フラッシュメモリ，ファイルサーバ，オンラインストレージなどは冗長性や分散性，セキュリティなどの問題が存在した．

そこで本研究は，Peer to Peer ネットワークを用いて冗長性や分散性，セキュリティなど問題の解決を図ろうとした．冗長性，分散性について本システムの Peer to Peer ネットワーク上のノード数を増やすことによって，セキュリティについてはデータの暗号化処理などを施すことによって他人からの閲覧を防ぐことで向上させる．

また，送信先に対してそのままのデータで送信してしまうと，送信先の負担が大きくなってしまう．この問題はファイルを二分割，更にパリティファイルというデータを生成し，それぞれネットワーク上の別のノードへ送信することで負担の軽減，ファイルの信頼性を図った．このシステムを用いれば地理的に離れている場所であっても容易にデータを分散することが出来る．

今後の展望としては，システムの実装や考察で述べたようなシステムを運用する上での問題点を解決する構想を練っていかなければならないことが挙げられる．

参考文献

[1] Van Bogart, John WC. Magnetic Tape Storage and Handling: A Guide for Libraries and Archives, Commission on Preservation and Access, 1400 16th St, NW, Suite 740, Washington, DC 20036-2217 (1995).

[2] Kim, Kwang-Kyu: Supporting device for minimizing vibration, noise and external impact of a hard disk drive, U.S. Patent No. 5, 587, 855, 24 Dec (1996).

[3] Adar, Eytan, and Bernardo A. Huberman: Free riding on Gnutella, First monday 5.10 (2000).

[4] 金子勇: Winny の技術, ASCII (2005).

[5] http://www.skype.com.

[6] SOFTIC 一般財団法人ソフトウェア情報センター: Napster 事件, http://www.softic.or.jp/lib/cases/riaa_v_napster.htm

[7] http://www2.accsjp.or.jp/criminal/2012/1199.php.

[8] 平野仁之, 岩切宗利, and 中村康弘: B-6-15 冗長度を付加する分散ストレージシステムの一実装方式, 電子情報通信学会ソサイエティ大会講演論文集 2003.2 15 (2003).

[9] 平野仁之, and 中村康弘: 冗長度とアクセス制御を考慮した分散ストレージシステムの一方式, 情報処理学会研究報告, CSEC,[コンピュータセキュリティ] 2003.126 53-57 (2003).

[10] Feldhofer, Martin, Sandra Dominikus, and Johannes Wolkerstorfer: Strong authentication for RFID systems using the AES algorithm, Cryptographic Hardware and Embedded Systems-CHES 2004, Springer Berlin Heidelberg, 357-370 (2004).

[11] Stoica, Ion, et al: Chord: A scalable peer-to-peer lookup service for internet applications, ACM SIGCOMM Computer Communication Review 31.4 149-160 (2001).