2008－DPS－137 (10)
2008／11／28

モバイルエージェントを基盤とした セキュアなP2Pファイル配信システムの開発

吉浦陽介† 長田智和††
谷口祐治††† 玉城史朗††

概要:近年，従来のクライアントサーバーシステムに代わり，さまざまなP2Pシステムが利用されている．P2Pシステムは技術自体は非常に有用だが，その性質上，セキュリティや著作権の面で大きな問題を抱えている．P2Pアプリケーションの代表的な例としてWinny[1]が挙げられるが，Winnyネットワーク上には，実際に音楽や映画などの著作権侵害の恐れのあるファイルや悪意のあるマルウェアが混入したファイルが多く流通している．そこで本研究では，P2Pシステムの安全性をより高いものとするモバイルエージェントを導入したP2Pファイル配信システムを提案する．また，それを，モバイルエージェント開発フレームワーク (AgentSpace) に実装し，その有効性を検証する．
キーワード:ファイル配信システム，P2P，モバイルエージェント

Study on Secure P2P File Delivery System based on Mobile Agent

YOSUKE YOSHIURA,† TOMOKAZU NAGATA,††
YUJI TANIGUTHI††† and SHIRO TAMAKI††

abstract:In recent years, P2P various systems are used instead of the conventional client server system. Although the technology of P2P system itself is very useful, it has the problem in respect of security or copyright.In the typical P2P application Winny, the file with fear of copyright infringement, such as music and a movie, and the file in which the malware was mixed are circulating.Therefore, in this study, the P2P file distribution system which introduced the mobile agent who makes the safety of P2P system higher is proposed.Moreover, the availability of the proposal system implemented using the mobile agent development framework (AgentSpace) is verified.
Key Words:file delivery system, P2P, mobile agent

1. はじめに

近年，インターネットが急速に普及してきている．それに伴い，インターネット上でのコンテンツの形態やサイズは多様化してきている．そのため，従来のようなクライアントサーバーシステムでの管理では，サーバーに大きな負荷やコストがかかってしまい管理効率が悪くなってしまう．また，コンピューター1台の性能が大幅に向上し，各コンピューター上で負荷の大きな処理が行えるようになってきた．こういった背景から，従来のクライアントサーバーシステムに代わり，コンテンツの分散管理が可能なP2Pシステムが注目されている[2]．現在では，さまざまなP2Pシステムが多くのユーザーに使用されている (2004年時点で，インターネット上の上りトラフィック全体の61%がP2Pトラフィックである)．[3]

P2Pシステムには，多くの利点が存在ある．コンテンツ管理にP2Pシステムを用いることによって，サーバーレスな環境を築くことが可能になるためサーバーに起因するコスト (運用・保守) を削減できるだけでなく，サーバーレスであるため，高いスケーラビリティを持つ．また，P2Pシステムは分散処理であるため，各コンピューター上での負荷を軽減することができる．さらに，ネットワークに接続されているコンピューターの1台に障害が起きた場合でも，処理を他のコンピューターに渡すことで，ネットワーク全体に及ぼす影響を最小限に留めることができ，高い耐障害

† 琉球大学大学院 理工学研究科 情報工学専攻
Engineering and Science,
Information Engineering Course,
Graduate School of the Ryukyus
E-mail:ysuke@neo.ie. u-ryukyu.ac.jp
†† 琉球大学 工学部 情報工学科
Department of Information Engineering,
University of the Ryukyus
E-mail:{nagayan,shiro}@ie.u-ryukyu.ac.jp
††† taniguchi@cc.u-ryukyu.ac.jp

性を実現することが可能である.[4]

上記で述べたように，P2Pシステムは技術自体は非常に有用だが，その性質上，セキュリティや著作権の面で大きな問題を抱えている．P2Pアプリケーションの代表的な例としてWinnyが挙げられるが，Winnyネットワーク上には，実際に音楽や映画などの著作権侵害の恐れのあるファイルや悪意のあるマルウェアが混入したファイルが多く流通している．また，P2Pシステムがサーバーレスで構築されることや，WinnyなどのP2Pアプリケーションにデータの回収・削除機能がないことが原因となり，一度データがP2Pネットワーク上に流出してしまうと，半永久的にP2Pネットワーク上を漂流してしまうことになる．そのため，P2Pアプリケーションの使用によって個人情報がP2Pネットワークに流出してしまい，被害を被っているユーザーも少なくない.[5]

そこで本研究では，P2Pシステムの安全性をより高めるため，モバイルエージェントを導入したP2Pシステムを提案する．モバイルエージェントもP2Pシステムと同じく，分散処理技術であるため，モバイルエージェントを用いることにより，P2Pシステムの利点を活かしつつシステムの改善が可能である．また，提案システムを，モバイルエージェント開発フレームワーク (AgentSpace) に実装し，その有効性を検証する．

2. 要素技術

2.1 P2P

P2Pとは，一般に，定まったクライアントおよびサーバーを持たず，不特定多数の個人間で直接情報のやり取りを行うコンピューターネットワークの形態，または，そのためのアプリケーションを指す．ネットワーク上の他のコンピューター (以下，ノード) に対してクライアントとしてもサーバーとしても働くようなノードの集合によって形成される[6]．また，P2Pは，ハイブリッドP2P(図1)とピュアP2P(図2)の2つのタイプに分類することができる．ハイブリッドP2Pは，ノードの管理は中央サーバーで行い，ファイルの転送は，ノード間で直接行う．ピュアP2Pでは，ノードの管理，ファイルの転送の両方を，ノード間で直接行う．

2.2 モバイルエージェント

モバイルエージェントとは，ネットワークを介した分散処理技術の一種である．ネットワークに接続されたノード間をエージェントと呼ばれるプログラムが移動しながら能動的に処理を行う.[7]

モバイルエージェントは自律的なプログラムであり，自律的に移動先を選択可能である．また，移動前のノード上でのプログラム実行状態 (変数内容など) やコードを一緒に移動先のノードに転送する．これによって，移動先では移動前の状態から処理を継続して行うことが可能であり，移動先のノードの計算リソースを利用することが可能である．さらに，モバイルエージェントは複製や永続化 (ファイル化) することができ，モバイルエージェント同士で通信することも可能である．

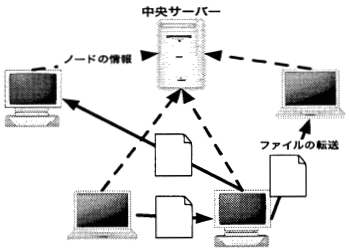

図1 ハイブリッド P2P

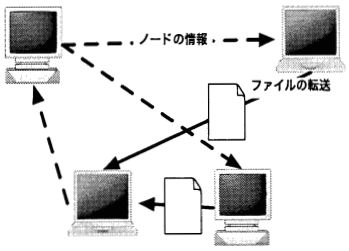

図2 ピュア P2P

モバイルエージェントが実行されるノードでは，モバイルエージェントの実行環境が備えられている必要があるが，実行環境が備えられていれば，パソコンや携帯電話，PDAなどあらゆる機器上で，モバイルエージェントは動作可能である．(図3)

以下にモバイルエージェントの利点をまとめる．

- ノード間通信の削減
  ノード間通信をノード内通信に局所化
- 非同期実行
  移動先ノードと移動元ノードは独立
- 通信切断対応
  移動後は通信が切断されても処理が可能
- 動的経路変更
  移動先と移動タイミングを自律的に決定・移動
- 並列実行・負荷分散
  複製を生成して並列処理が可能
- 通信回数・遅延の低減化
  通信相手側ノードに移動・処理
- プラットフォーム独立
  OS・ハードウェアに依存せず処理が可能

一方で，モバイルエージェントには，エージェント自体に対しての盗聴・改ざんをいかに防ぐかというセキュリティ問題がある．

2.3 AgentSpace

AgentSpaceは，Java言語を利用した分散計算用ミ

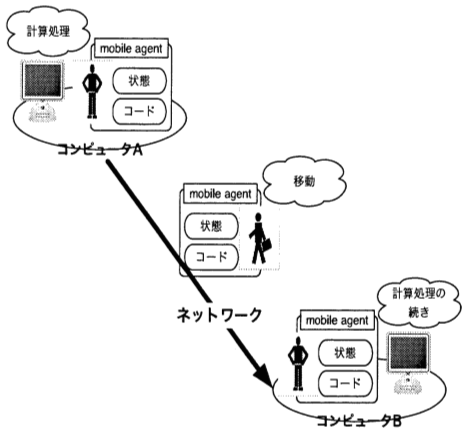

図 3 モバイルエージェントの動作

ドルウェアである[8]．モバイルエージェントの作成を可能にする API 群とそのエージェントの実行や移動などを処理するランタイムシステムの二つの部分から構成されている．(図 4)

2.3.1 エージェントランタイムシステム

エージェントランタイムシステムは Java の仮想機械 (VM) 上で動作し，エージェントの起動，永続化，停止を管理する．また，他のエージェントランタイムとのエージェントの送受信を行うことが可能であり，エージェントを受信するとそのエージェントを動作可能にし，転送要求のあったエージェントを他のエージェントランタイムに転送する．

エージェントランタイムは，各エージェントとのインタフェースとなるエージェントコンテキストと，エージェント移動や永続化を管理するエージェントサーバーから構成されている．

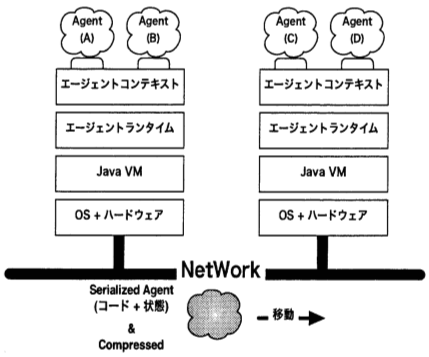

図 4 AgentSpace システムアーキテクチャ

2.3.2 関連システムとの比較

AgentSpace は，Java ベースのモバイルエージェントシステムである．IBM 社の Aglets[9]，Fujitsu 社の Kafka[10] などとは類似性が見られる．しかし，他のシステムとの重要な相違点として，エージェントの転送方法が挙げられる．具体的には，他の Java ベースのモーバイルエージェントシステムでは，実行状態を転送した後，必要なコードをオンデマンドで転送するため，転送するデータ総量は少くなる可能性がある．多くのモバイルエージェントシステムでは，1 つのコードを転送するのに最小でも 2 回の通信を必要とするため，エージェント転送に時間を要する原因となっている．AgentSpace では，1 つのエージェントを構成する全ての情報をひとまとめにして転送する方法をとっているため，通信回数を削減し，高速化を図っている．

3. 提案システム

3.1 概　　要

1 章で述べた通り，P2P システムを使用してやり取りされているファイルの中には，内容が改ざんされているものや，悪質なマルウェアが仕込まれたものが存在する．そのため，モバイルエージェントをファイルと一緒に送信することで，モバイルエージェントにファイルの検査を任せる．さらに，より安全性を高めるために，ファイルを送信する際は暗号化して送信し，この暗号化 (復号化) 作業もモバイルエージェントに担わせる．また，P2P システムによるファイル送受信では，そのファイルの権利保護に関する問題が付いて回る．そこで，ファイルの辿った通信経路やファイルに対するアクセル回数およびコピー回数などの情報をモバイルエージェントにログとして収集させ，送信ファイルの管理・監視を行える機能を付加する．

本研究では，提案システムを，モバイルエージェントの研究開発用フレームワークである AgentSpace に実装した．2.3 節で述べるが，AgentSpace は Java ベースのシステムであるためプラットフォームに依存しないシステムの開発を行うことができる．

3.2 ファイルの検査

提案システムでは，最初にファイルを送信する際に，モバイルエージェントがファイルの各種情報を取得し，保持する．ファイルが他ノードで受信されるとそのノード上でモバイルエージェントは再びファイルの情報を取得し，最初に取得した情報と比較して，ファイルの改ざん・マルウェアの混入などを検査する．その後，モバイルエージェントは新しいノードに移動する度に同じ動作を繰り返す．

今回，モバイルエージェントにがファイルの検査に利用する情報として，以下の情報を用いた．

- ファイル名
  送信前と受信後でファイル名に変化がないか検査

する．

- ファイルサイズ
  送信前と受信後でファイルサイズに変化がないか検査する．
- ファイルのハッシュ値 (メッセージダイジェスト)
  送信前と受信後でファイルのハッシュ値に変化がないか検査する (ハッシュ関数は SHA-1[11])．
- ファイルの拡張子
  送信前と受信後でファイルの拡張子に変化がないか検査する．また，ファイルの拡張子と実際のファイルタイプが一致しているかを検査する．
  ここで，ファイル拡張子はプログラムによって簡単に得ることができるが，拡張子を得るだけでは，送信前と受信後の変化は検出できるものの，実際のファイルタイプ (ファイル拡張子の偽装) を検出することはできない．そこで，今回は，*URLConnection* クラスの *guessContentTypeFromStream* メソッドを使用することにした．*guessContentTypeFromStream* メソッドは，対象ファイルをバイナリデータとして読み込み，解析することで，実際のファイルタイプを検出することが可能になる (各ファイルはファイルタイプにより，固有のヘッダ情報を保持している．このヘッダ情報を解析することにより，ファイルタイプを検出することができる)．

3.3 ファイルの暗号化・復号化

3.2 節における提案システムでファイルの検査について説明したが，より安全性を高めるためには，ファイルの改ざん・マルウェアの混入を事前に防ぐことが重要である．そこで，ファイルをノード間でやり取りする際には，ファイルを暗号化することとした．モバイルエージェントは，ファイルを送信する際，ファイルを暗号化した上で，他ノードに送信する．また，他ノードで暗号化されたファイルを受信すると，再びモバイルエージェントによって復号化する． 暗号化アルゴリズムには AES 暗号を使用した．また，暗号化されたファイルには「 *~.aes*」という拡張子を明示的に付加するようにした．

3.3.1 AES 暗号

AES 暗号は，2001 年に制定された米国の標準ブロック暗号 (共通鍵暗号) である．以下に概略を示す．

AES 暗号の鍵長は，128，196，256 ビットの 3 通りから選ぶことができる (提案システムでは 128 ビットを使用している)．4 バイトを 1 ワードとし，鍵 K のワード長を N_k で表す．従って，$N_k = 4, 6, 8$ である．暗号化における繰り返しの回数 (ラウンド数) を N_r で表す．$N_r = 10, 12, 14$ である．それぞれの鍵長 N_k に対し，ラウンド数 N_r が表 1 のように規定されている．また，AES 暗号のブロック長は，128 ビット (16 バイト) である．

表 1 鍵長とラウンド数

鍵のビット長 [ビット]	128	196	256
鍵のワード長 N_k[ワード]	4	6	8
ラウンド数 N_r	10	12	14

3.3.2 暗号化・復号化の手順

モバイルエージェントは，入力としてファイル名とパスワードを受け取る．受け取ったパスワードは鍵の種としてハッシュ関数 SHA256 に通され，鍵が生成される．生成された鍵によって入力されたファイルの暗号化・復号化を行う (SHA256 は 256 ビットのハッシュ値を生成し，AES の鍵長は 128 ビットである．鍵長よりもハッシュ値の方が長いため，生成される鍵の組み合わせが増える．そのため生成された鍵は，衝突するリスクを軽減することができる)．

ファイルの暗号化・復号化処理はファイルの拡張子「 *~.aes*」によりモバイルエージェントが判断する．ファイルの拡張子が *~.aes* 以外の場合は未暗号化ファイルのため暗号化処理を行い，ファイルの拡張子が *~.aes* の場合は，暗号化されたファイルであるため復号化処理を行う (図 5)．

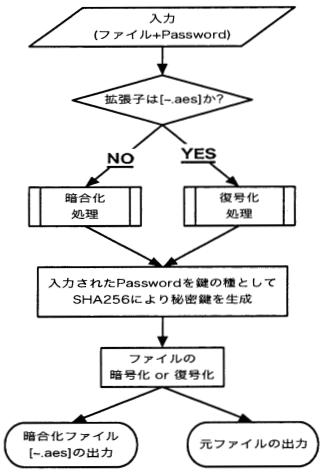

図 5 暗号化・復号化フローチャート

3.4 ファイルの管理・監視

本研究では，ファイルの管理・監視を行う方法として，ファイルの送受信可能な回数 (コピー回数) に制限を設定することにした．また，ファイルの通った経路上の全てのノードの情報をモバイルエージェントに取得・保持させることでファイルの回収を行い，ファイルの拡散を防ぐことを可能にする．

以下でファイルの管理・監視機能の具体的な説明をする．

- ファイル送信時にコピー回数制限を設定

ファイルの送信者が，ファイルの送信前に，ファイルに対してコピー回数制限を設定することができる．設定されたコピー回数制限は，送受信の度にモバイルエージェントにチェックされる．コピー回数制限を超えるとそれ以上のファイルの転送は不可能となり，ファイルは破棄される．

- ファイルを受信したノードの情報をモバイルエージェントが取得・保持
  モバイルエージェントは，ファイルを受信したノード全てに対して，ノード情報を取得・保持する．送信側ノードは，モバイルエージェンに対してこの情報を要求し，閲覧することで，ファイルの辿った経路を知ることができ，ファイルの管理・監視に活用することができる．
- 送信ファイルの削除 (回収)
  上記で述べた各ノードの情報を用いることによって，ファイルの回収を行えるようにした．ノードの情報の内，IP アドレスと受信したファイルの各情報 (名前，ハッシュ値) をモバイルエージェントに渡す．モバイルエージェントは，指定されたノードに移動し，対象となっているファイルを検索する．検索条件に適合したファイルを検出するとそのファイルはモバイルエージェントによって削除される．

3.5 システム全体図

3.2 節 ~3.4 節で述べた機能を実装した提案システムの全体図を図 6 に示す．

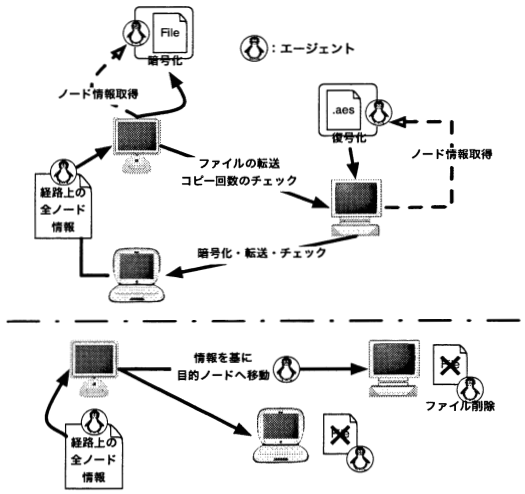

図 6 提案システムの全体図

4. 評価・考察

提案システムを評価するにあたり，P2P システムを使用する上で実際に起こりうるセキュリティや著作権保護に関する問題を想定した．その上で，これらの問題が提案システム上でいかにして回避されるかを実装した機能ごとに検証した．

4.1 ファイルの検査

ファイルの検査機能を実際に検証するために，ファイル名変更・ファイル内容変更・ファイル拡張子変更をそれぞれ組み合わせて行った上で，ファイルの送受信を行った．

3.2 節で挙げた 4 種類の情報のうち，1 種類の情報のみを用いる場合だと，ファイルの改ざんを見落としてしまう場合があった．しかし，4 種類の情報全てを用いて総合的に受信したファイルの検査を行うことで，送受信の間にファイルに対して何らかの操作・改ざんが行われたことを受信したファイルを開く前に，検出可能であることを確認した．

4.2 ファイルの暗号化・復号化

ファイルをモバイルエージェントによって，暗号化して送受信することにより，第三者によるファイルの盗聴・改ざんに対して高い信頼性を保つことができた．また，仮に第三者にネットワーク上で不正に暗号化ファイルを取得され，改ざんされたとしても，改ざんされた暗号化ファイルは，正常には復号化されない．そのため，3.2 節で述べたファイルの検査機能により，受信したファイルの改ざんを容易に検出することができる．

4.3 ファイルの管理・監視

ファイルの管理・監視機能の 1 つとして，コピー回数制限を設定できるようにしたことで，1 次コピーファイルの拡散を最小限に留めることができた．また，提案システムでは，ファイルの送信者がコピー回数制限を自由に設定することが可能である．そのため，既存のコピーワンスやコピーテンスといった DRM 技術とは違い，自由度の高いシステムとなっている．

さらに，モバイルエージェントがファイルの辿ったノードの情報を取得・保持し，ファイルの送信者がその情報を閲覧することで，ファイルの経路を監視することができた．また，この情報をもとに各ノード上で受信されたファイルを削除し，回収することができた．これによって，誤って送信してしまったファイルや著作権侵害の恐れのあるファイルなどがネットワーク上で，ネズミ算式に増えていくことを防ぐことができた (ファイルのハッシュ値を検証するため誤って他のファイルを削除してしまうことはない)．

以上のように，モバイルエージェントを用いることによって，上記の操作はモバイルエージェントが自律的に判断して実行するため，ユーザーは従来の P2P システムと違和感なく使用することができると考えられる．

5. 総　　括

本研究では，モバイルエージェントを基盤として，P2P システムを開発することで，安全性の高いシステムを実装することができた．また，全ての機能をモバイルエージェントに実装することによって，1 つのシステム上で，包括的なセキュリティ・著作権保護対策を行うことができた．さらに，モバイルエージェントを用いて開発を行ったことによって，従来の P2P システムと比べてシームレスなシステムにすることができた．提案システムでは，主な機能として，3.2 節~3.4 節の 3 種類の機能を実装したが，機能のアップデートや追加を行う際は，モバイルエージェントの機能としてアップデート・追加していくことができるため，拡張性・汎用性という点でも高い性能を持たせることができた．

5.1 今後の課題

モバイルエージェントを使用する際には，2.2 節でも述べたように，エージェント自体に対する盗聴・改ざんをいかにして防ぐかが最大の課題となる．提案システムでは，ハッシュにより改ざんを検査したり，暗号化により盗聴を防ぐ機能を実装したが，これらの機能だけで十分であるかの検討およびさらなる改善の必要がある．また，3.2 節のファイルの拡張子の検査方法の部分で述べた *guessContentTypeFormStream* メソッドは，対応している拡張子の数がまだ不十分であり，精度も完ぺきといえるものではないため，拡張子の検査には，*guessContentTypeFormStream* メソッドのさらなる改良・アップデートが不可欠である．さらに，今回の提案システムで実装したファイルの管理・監視機能は，1 次コピーと 1 次コピーファイルの拡散を抑止することができるが，2 次・3 次コピーとなるとそのファイルの管理・監視は困難である．そのため，2 次・3 次コピーをどのように防ぐかも今後の重要な課題の 1 つである．

最後に，パケットのフィルタリング機能などさらなる機能をモバイルエージェントに実装することで，より安全性を高めることができる考えられる．

参　考　文　献

1) 金子 勇: ”Winny の技術”，アスキー書籍編集部，2005
2) ARIEL NETWORKS: ”P2P 概論-ありえるえりあ-”，http://dev.ariel-networks.com/articles/unixmagazine/abstract/
3) TorrentFreak: ”’Shocking’ 61% of all Upstream Internet Traffic is P2P”，http://torrentfreak.com/shocking-61-of-all-upstream-internet-traffic-is-p2p-081021/
4) 横澤 誠: ”P2P アーキテクチャの可能性”，野村総合研究所，2002
5) INTERNET Watch: ”インターネット 事件簿”，http://internet.watch.impress.co.jp/static/column/jiken/2004/05/27/
6) higaitaisaku.com: ”P2P について”，http://www.higaitaisaku.com/
7) 佐藤 一郎: ”モバイルエージェント”，国立情報学研究所，2002
8) モバイルエージェントシステム: ”AgentSpace”，http://research.nii.ac.jp/ichiro/agent/agentspace.html
9) IBM 東京基礎研究所: ”Aglets について”，http://www.trl.ibm.com/aglets/about.htm
10) T. Nishigaya: ”Design of Multi-Agent Programming Libraries for Java”，http://blacky.fujitsu.co.jp/hypertext/free/kafka/paper/
11) 黒澤 馨，尾形 わかは: ”現代暗号の基礎数理”，コロナ社，2004
12) フリー百科事典ウィキペディア: ”Wikipedia”，http://ja.wikipedia.org/