

Winny の仕組み

3

● 章 ●

Winny は、複数のノードが協調しながらファイル共有を実現しています。この章では、最も単純なファイルの流通を例に、公開されたファイルが他のユーザーにダウンロードされるまでの仕組みを追い、どのようにして匿名性のある効率のよいファイル配布が可能になったかを解説します。

3.1 Winny ネットワークの概観

Winny はピュア P2P 型のシステムなので、中心的な役割をはたす特定のサーバは存在しません。また、P2P ネットワーク内のノードやファイルすべてを把握しているノードもありません。近隣のいくつかのノードだけを知っているノードがインターネットを使って多数接続し、他のノードの情報やファイルの所在を交換しながらファイルの検索や転送を可能にしているのです(図 3-1)。

各ノードがインターネットを介して接続し、形成しているネットワークを、Winny ネットワークと呼ぶことにします。Winny ネットワークは、「インターネット上に Winny ノード相互が築いたアプリケーションによるネットワーク」だといえます。各ノードは利用者の都合によって起動したり停止したりするので、Winny ネットワークは全体としてノードの参加と離脱がかなり頻繁な、動的なネットワークになります。このため Winny の設計も、ノードがいつ離脱してもネットワーク全体の機能に大きく影響しないようになっています。

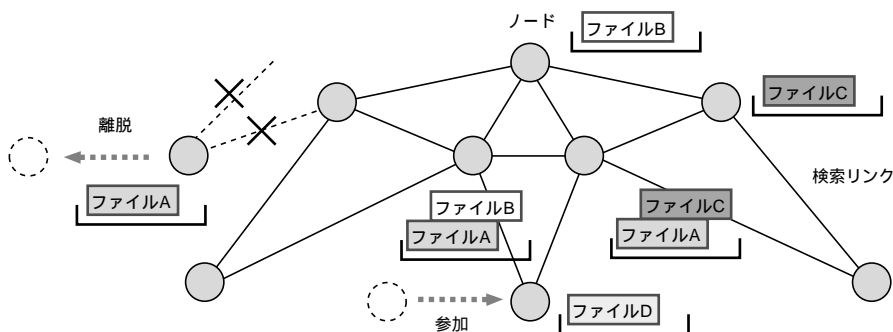


図 3-1 Winny ネットワーク

以降では、図 3-2 のような簡単な Winny ネットワークを使って説明します。

Winny を起動すると、そのノードは Winny ネットワークに接続し、終了するまで常に接続を維持します*1。Winny は、このノード間の接続(コネクション)を通じて、他のノード情報やファイルのキーを交換したりファイルを検索したりするので、この接続のことを「検索リンク」と呼びます。なお、ファイルを転送するときには、ファイルの転送元ノードと転送先ノードとの間でこれとは別の接続を開きます。転送に使用する接続を「転送リンク」と呼びます。

検索や配送の効率を高めるため、Winny は高速な回線に接続しているノードにより多くのキーやファイルを集めるようになっていきます。このために、Winny ネットワークには、高速な回線に接続するノードを上流、低速な回線に接続するノードを下流とする、特有の概念があります。Winny ネットワーク内のノードを大まかに分類すれば、FTTH に接続している高速回線のノード、ADSL に接続しているノード、ISDN に接続した低速回線のノードがあり、高速回線ほど他のノードにサービスする能力が高いと考えられます。この概念にどのような効果があるかは、以降で明らかになっていきます。

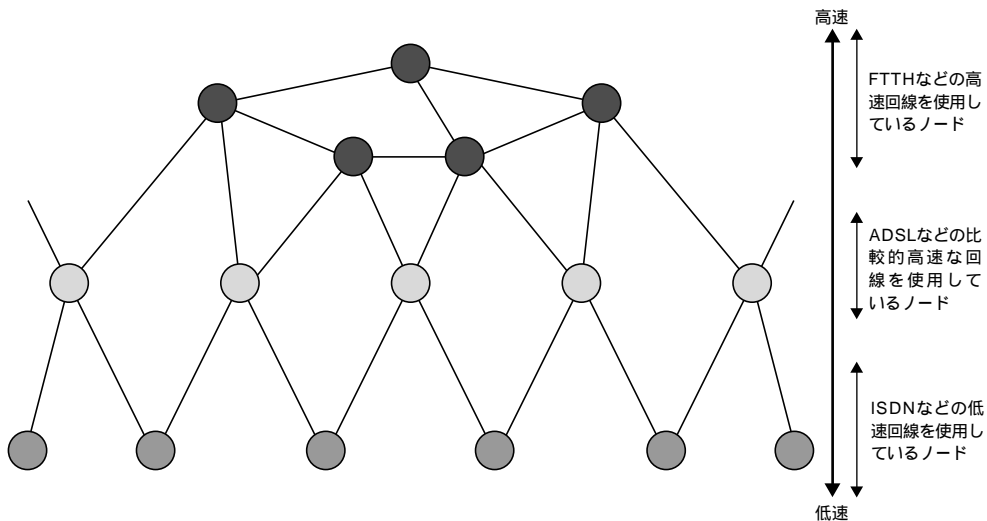


図 3-2 この章で使う Winny ネットワークの例

*1 Winny ノードは、起動時に上流(なければ自分と同等)のノード 2 つに接続を要求し、Winny ネットワークに参加したあとは、下流(もしくは自分と同等)のノード 5 つまでの接続要求を受け付けます。詳しくは 4 章「実装」をご覧ください。

3.2 ファイルの公開からダウンロードまで

以降では、公開されたファイルがファイルを探しているユーザーに届けられるまでの過程を、順に追っていきます。

ファイルの公開

ユーザーが、公開するファイルをアップロードフォルダに置くと、Winny はそのファイルからキーと本体(ボディ)を作成します。キーは、いわばファイルの要約情報であり、ファイルの名前や大きさ、更新時刻、ファイルの識別に使うファイル ID (ハッシュ値)、ファイル本体の位置情報を示す IP アドレスとポート番号などが含まれています。ファイル本体のほうはファイルの内容そのものです。

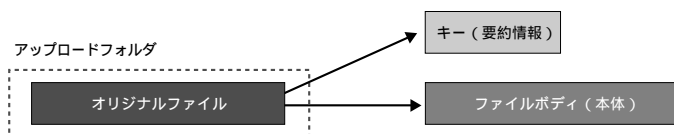


図 3-3 ファイルのキーと本体

Winny は、検索リンクでつながっている隣接ノードとのあいだで、持っているキーを定期的に変換するので、アップロードフォルダのファイルのキーはしだいに Winny ネットワークに拡散していきます。

🔗 ハッシュ値

ファイル ID となっているハッシュ値は、ファイルの内容全体から数学的なアルゴリズムで算出した一定サイズのデータです。ハッシュ値には、元のデータが異なればおおむね重複のない一意な数値になるという性質があるので、ファイルの内容の要約情報と考えることができます。Winny ではこれを利用して、ファイルの内容に応じた一意の ID を付けています。つまり、たとえファイル名が異なってもファイル ID が同じであれば同じ内容であり、ファイル ID が異なれば別の内容として扱います^{*2}。

*2 ファイルの内容からハッシュ値への計算に使う手順をハッシュ関数と呼びます。Winny では、128 ビットの MD5 をハッシュ関数として使っています。

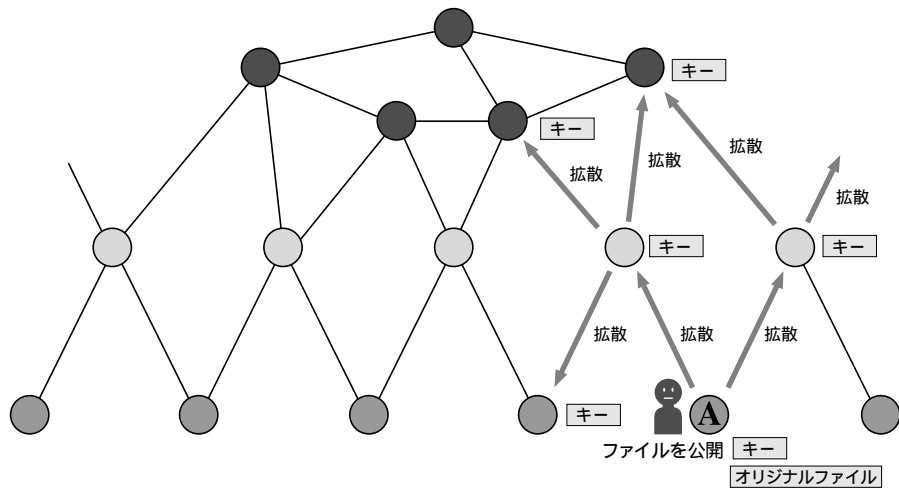


図 3-4 ファイルの公開とキーの拡散

図 3-4 は、ノード A がファイルを公開したときのような様子です。

各ノードが保有するキーは、ノード間の定期的な交換と、次節で説明する検索によって、時間の経過とともに更新されていきます。

キーの寿命とリフレッシュ

このように、Winny の各ノードにはキーが集まってきますが、キーには寿命があります。他のノードから運ばれてきたキーは、一定の時間がたつと有効期限が切れ、キーのリストから削除されます。ファイルを公開しているノードが稼働しているかぎり、そのファイルのキーは定期的に運ばれてくるので、たとえ古いキーが消滅しても新しいキーの到着でキーは延命します。しかし、ファイルを公開しているノードがネットワークから切り離されれば、対応するキーは一定時間を経過したあと各ノードのキーのリストから削除されます。つまり、ファイルを持つノードがなくなればキーも消滅し、他のノードが検索をしても見つからなくなるというわけです。

ファイルの検索

公開ファイルのキーが拡散したところで、別のノードがこのファイルに関連するキーワードを指定して検索したとしましょう。Winny は、検索用のクエリ(問い合

わせ)パケットに検索条件の文字列を詰め、上流のノードに送り出します(図 3-5)。

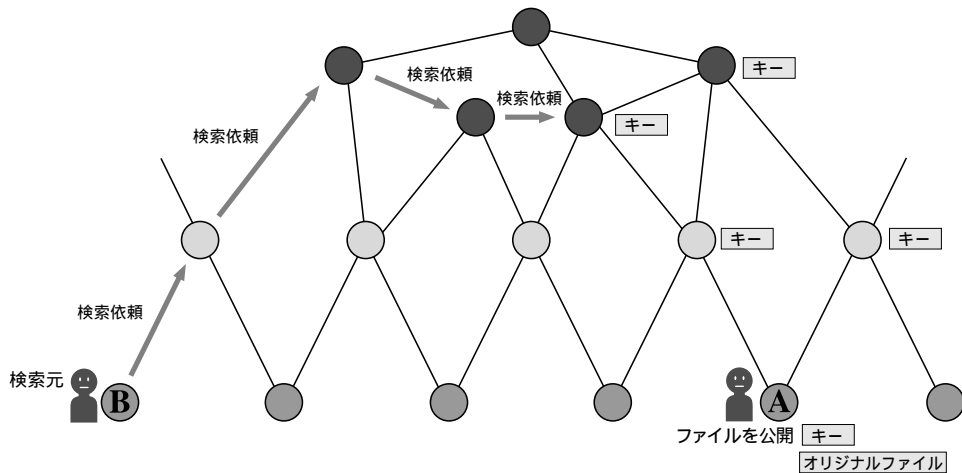


図 3-5 ファイルの検索

クエリは、転送先のノードで条件に合うキーがあるとそれを詰め込まれ、さらに隣接するノードへと転送されます。何度も転送を繰り返したところで、往路を逆にたどって検索元に送り返されます(図 3-6)。

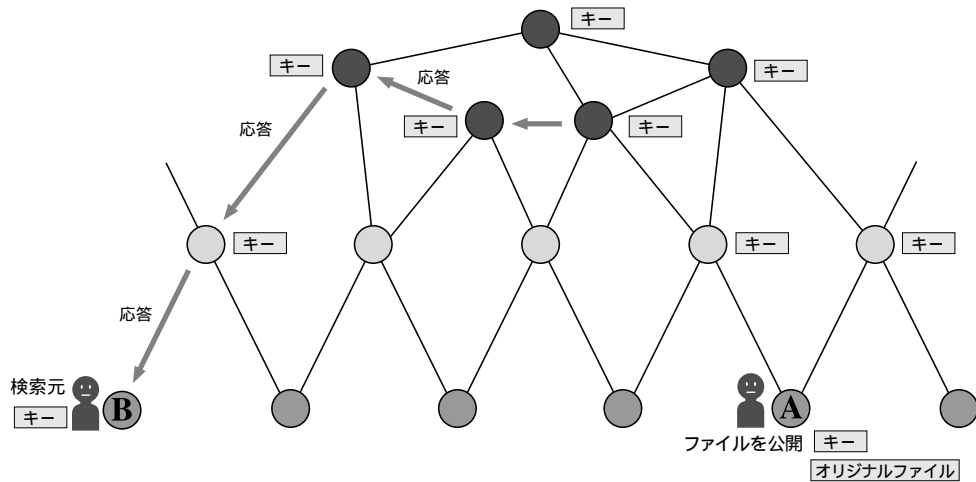


図 3-6 キーの入手

検索元ノードは、経由したノードで見つけたキーのリストを受け取ります(もちろん、不幸にして条件に合うキーが見つからない場合もあるでしょう)。このようにして検索元のノードは、隣接ノードとの定期的なキーの交換を待たずにクエリという積極的な方法を使って、遠方のノードにあるキーを検索し、キーを手に入れることができます。

検索を上流に向かって転送することには、重要な意味があります。まず、クエリの経路となる上流ノードにキーが拡散されるので、キーはしだいに上流ノードに集まっていきます。こうなったあとは、上流方向を検索するだけで、効率よく目当てのキーを見つけることができるようになります。

検索にはもうひとつ、クエリの通り道となったノードにも収集したキーが複製されるという効果があります。すなわち、キーの拡散は、隣接ノードとの定期的なキー交換と検索の2つのタイミングで行われます。

ファイルの転送

さて、定期的なキーの受信と検索操作でノードに集まったキーは、Winnyの検索画面に一覧表示されています。ユーザーがそのなかから適当なものを選んでダウン

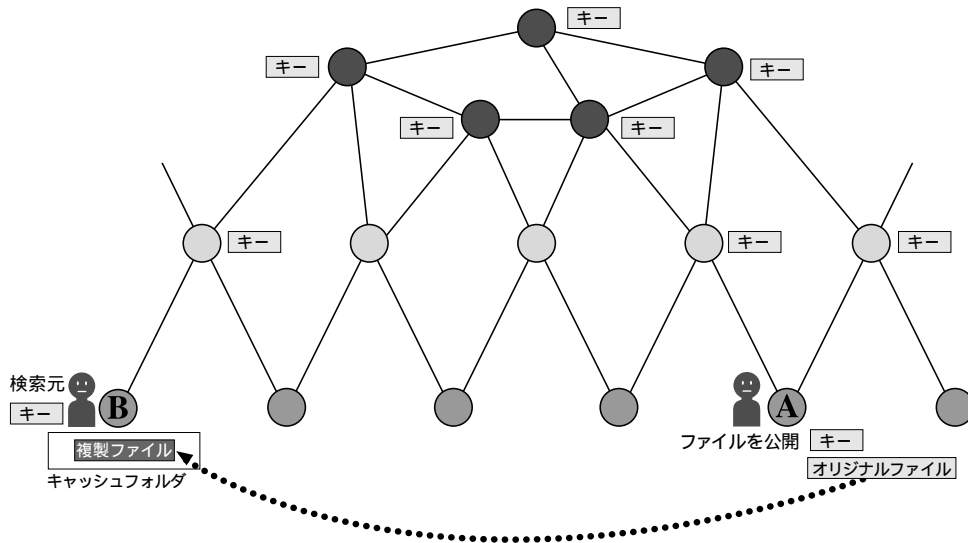


図 3-7 ファイルの転送

ロードを指示すると、ファイルの転送が始まります。検索結果のキーには、ファイル本体を持っているノードの情報(IP アドレスとポート番号)が含まれているので、Winny はそのノードと接続し(新たなコネクションを張る)、ファイルの転送に使用します(図 3-7)。このコネクションを転送リンクと呼び、ファイルを1つ転送するごとに開き、転送が終わると閉じます。

実際には、ファイルは暗号化したキャッシュファイル形式で転送され、ダウンロード側ノードのキャッシュフォルダに格納されます。転送が完了すると、ファイルはキャッシュファイル形式から元どおりのオリジナルファイルに復元され、ダウンロードフォルダに置かれます。

ダウンロードしたファイルは公開される

実は、ファイルをダウンロードしたあと、キャッシュフォルダに残ったファイルも公開され、他のノードから検索したりダウンロードしたりできるようになります。このことは、これまで説明してきた Winny の匿名性の実現方法とも関連してきますが、これについてはあとで詳しく説明します。

図 3-8 のノード B には、キャッシュしたファイルに対応するキーが作成され、隣接ノードとの定期的なキー交換で拡散していきます。

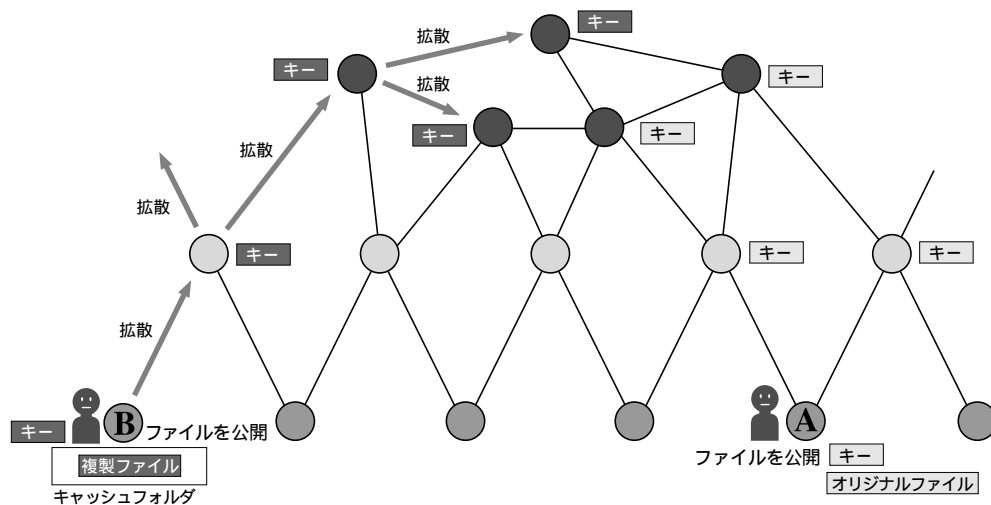


図 3-8 キャッシュの公開