平成22年度卒業論文

論文題目

匿名経路制御に適した
シミュレーションツールの提案

神奈川大学 工学部 電子情報フロンティア学科
学籍番号　200702957
原木　章成

指導担当者　木下宏揚　教授

目 次

図 目 次

第1章　序論

近年PCの高機能化やネットワークの高速化によりインターネットへのアクセスが自然と行われるようになってきた．さらにインターネットでのコンテンツ配信などのサービスが新たに開始されたことによって，PCを利用した音楽，動画，画像などのデジタルコンテンツの流通が盛んになってきている．[1] デジタルコンテンツの最大の特徴として，無劣化で複製が容易であるということがあげられる．[2] これよりP2Pネットワーク等を介し，無断送信が禁止されているDRMの目的はコンテンツの不正コピーや不正流通の防止であるが，単なる不正防止というばかりでなく，従来の枠組みにとらわれない新たなコンテンツの流通を実現するためにも欠かせない技術となっている．本来，DRMをうまく活用することでコンテンツ流通が持つ様々な課題は，改善するこは技術的に可能である．しかしながら，各流通メディアやサービスでDRMが異なっていることによる互換性の欠如や恒久的な再生が保障されていないこと，私的利用等の合法的使用の際にまで消費者の権利に対する不当な制限をあたえてしまっているのが現状であり，DRMに対する批判の声すら上がっている．[3] 恒久的な再生が保障されていないなど改善すべき点が多いこれらを改善するため要素技術の一つとして匿名通信とその経路制御が必要となってくる．しかし，従来のネットワークシミュレーションでは使用目的が異なるため困難であった．従来の方法ではコンテンツの配布や譲渡によって移動するコンテンツを保持するノードを追跡可能であったためコンテンツの権利者への匿名経路制御出来ることが必要であった．そこで匿名経路に適したシミュレーションツールを提案することを目的とする．

第2章　基礎知識

2.1　著作権

2.1.1　著作権概要

著作権は特許権、商業権などの産業財産権とともに「知的財産権」と呼ばれる権利の一つである．産業財産権が産業経済の発展を目的としている制度であるのに対し,「著作権」は文化の発展を目的とし,音楽，絵画，小説，映画，コンピュータ・プログラムなどの著作物を保護することを目的としている．[8] 著作権法によると,「著作物とは,思想又は感情を創作的に表現したものであって，文学，学術，美術又は音楽の範囲に属するものであり，著作権はそれを創作する者を指す」と定義している．また，著作者の権利は人格的な利益を保護する著作者人格権と財産的な利益を保護する著作権（財産権）の二つに分かれる．著作者人格権と著作権（財産権）は，それぞれ様々な権利から成り立っているが，DRMおよびコンテンツの流通を考える上で特に重要となるのは，著作者人格権の中の「複製権と送信可能化権を含む「公衆送信権」である．[9] 著作権法では，著作物の利用者は私的使用の為の複製が認められている（第30条）．ただし，私的使用の為の複製は原則的に利用者が自身の機器を用いて自分自身で行う必要があり，また，技術的保護を回避してまで行うことは認められていないため，私的使用の為に複製した著作権を勝手にインターネット上に公開したり，P2Pなどのファイル交換ソフトを利用して不特定多数に提供してはならない．[7] 著作権に対する理解と保護の度合は，その国の文化のバロメータと言われている．社会の中で著作権が果たしている大切な役割を尊重し，著作物を利用する際に，著作者への正当な対価を支払うことが，さらなる著作の文化を

産み，文化を発展させていくことにつながる．

2.1.2 権利記述言語

ディジタルコンテンツの保護に関する権利と条件を規定する為には，それらの情報はコンピュータが翻訳可能な言語で記述する必要がある．それは，単なるフラグで表現されることもあれば．権利記述言語（REL : Rigths Expression Language）を用いて記述されることもある．権利記述言語の標準化動向としては，XML ベースの仕様である XrML（eXtenible Markup Language）や ODRL（Open Digital Rights Language）があげられる．権利記述言語は文法的に明確な定義を持つ言語であり．条件指定の仕方次第で，かなり複雑な権利情報を記述することが可能である．

2.2 DRM（ディジタル著作権管理）

ディジタルデータとして表現されたコンテンツの著作権を保護しその利用や複製を制御・制限する技術．主な技術としては音楽，映像ファイルにかけられる複製の制限や電子透かし，iTunes における FairPlay，Adobe，LifeCycle などがあげられる．ディジタル化されたコンテンツは何回でもコピーしても品質が劣化しないため P2P など違法な配布・交換増えているこれに対抗するためにコンテンツの流通・再生に制限を加える DRM 技術が注目を集めている．[5] DRM はコンテンツ利用者の利便性を損なうことなく著作権および所有者に適切な対価を還元することを目標としている．DRM を実現する物には様々あり，その機構はコンテンツの形式や利用形態によって異なるがユーザが特定のソフトウェアを使い，暗号化されたコンテンツを復号しながら再生する方式が一般的である．暗号化に使われる鍵は再生ソフトウェア内に隠されているか，あるいはネットワーク上からダウンロードされることが多い．この再生ソフトウェアがユーザのコンテンツ用を管理するため，利用期間の切れた後には再生不可能にするなどの処置が可能になる．しかしながら，DRM 以下の理由から批判の声が上がっているのも現状である。[3]

- 恒久な再生が保障されていない
  DRM 技術のほとんどが特定メーカーによって定められ，その技術的詳細が一般に公開されないことから，そのメーカーやサービスが活動を停止した際に，購入したコンテンツが将来にわたって利用可能なのかが必ずしも担保されない．

- 消費者の権利に対する不当な制限
  DRM はその技術的特性により，通常，複製以外の利用（著作権法によって認められている範囲での抜粋や，他人への譲渡など）も制限することが多い．このため，DRM は購入した製品を自由に使う消費者の権利を奪ってるとの主張もある．

2.3 情報カプセル

2.3.1 カプセル化

カプセル化とは，オブジェクト指向プログラミングが持つ特徴の一つであり，データとそれを操作する手続きを一体化して「オブジェクト」として定義し，オブジェクト内の細かい仕様や構造を外部から隠蔽することである，[5] 外部からは公開された手続きを利用することでしかデータを操作できないようにすることで，個々のオブジェクトの独立性が高まる．カプセル化の利点としては以下のようなものが挙げられる [10]

- 不正な操作からの保護
- 複雑さの隠蔽
- 部品化/再利用性の向上
- 修正/変更に対する影響範囲の極小化
- バグの影響範囲の極小化

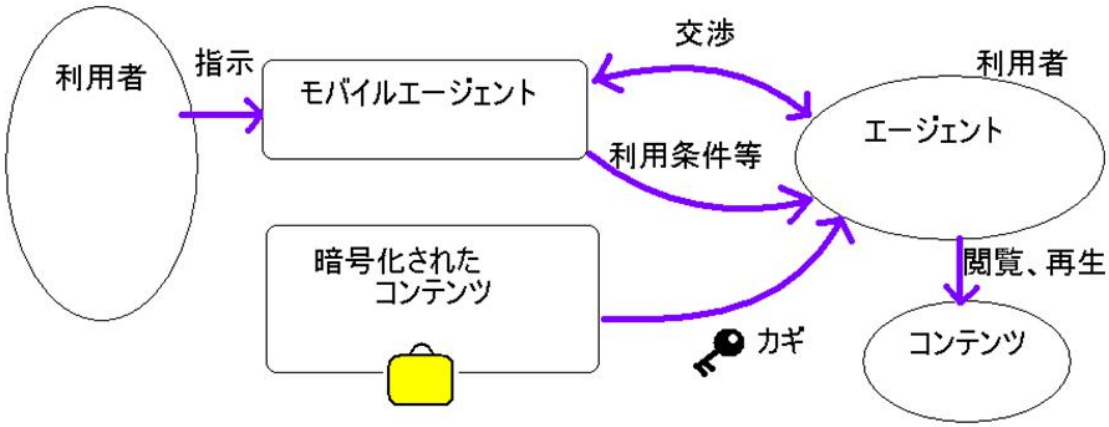

図 2.1: コンテンツの流れ

P2P ネットワークの普及などによるコンテンツの不正流通が社会問題なってい中，カプセル化はコンテンツを保護する手段として非常に有効である，
カプセル化はコンテンツを自体は超流通的に自由にコピー・転送され，電子鍵と組み合わせることによって，コンテンツの閲覧・再生が可能となるという仕組みにすることで実質的にコンテンツの権利を保護することが可能となる．[11]
図は情報カプセルのコンテンツの流れを表している．

2.3.2 情報カプセル

暗号化されたコンテンツは，情報カプセルエージェントにより管理されているため，利用条件が満たされない限り，コンテンツを復号し，再生することはできない．ただし，カプセル化コンテンツ自体はコンテンツ管理者のサーバー，P2P ネットワーク等を介し，自由に流通できるものとする．

図 2.2: カプセル化コンテンツ

2.4 エージェント

2.4.1 エージェントとは

普通名詞としての「エージェント」という言葉には「代理人」という意味がある．この意味を文字どおりに解釈すると，エージェントはユーザーの代理人として機能するシステムと考えることができる「エージェント」という用語はソフトウェアの抽象化・アイデア・概念を説明するものであり，その意味でオブジェクト指向プログラミングの各用語と同類である．また，様々な人々がそれぞれにエージェントの定義を提案しているが，それらには以下のような概念が含まれている．[11]

- 永続性
  そのコードは要求されて実行されるのではなく，常に起動された状態で，何らかの行動を起こす時期を自身で判断する．
- 自律性
  エージェントは，実行すべきタスクの選択優先順位を付け，目標に向けた行動，意思決定を人間の手助けなしで行う機能を持つ．
- 社会性
  エージェントは他のコンポーネントと何らかの通信や協調をする機能持ち，1つのタスクを共同で処理する．
- 反応性
  エージェントは周囲の環境を把握し，その変化に適切に反応する．

2.4.2 エージェントの分類

多くの場合，一つのエージェントがすべての機能を実現するのは難しい．また，エージェントという名称を使う場合には，ある機能に着目して，いろいろな形容詞をつけて呼ばれることが多い．[11]

- 自律エージェント
  自己充足的であり，観測された環境基づいて内部目標を達成するための行動を独自の判断で決定する．
- 知的エージェント
  一種のの人口知能的機能を有するエージェントで，ユーザーを補助し，繰り返し行うべきコンピュータ関連のタスクをユーザーの代って行う．
- マルチエージェント
  単体では目標を達成できず，複数のエージェントが相互作用を及ぼしながら動作する．
- モバイルエージェント
  ネットワークに接続されたコンピュータ間をプログラムが移動しながら処理を行う．

2.5 ルーティング

ルーティングとは，ネットワーク上でデータを送受信する際に，そのデータがきちんと相手先に届くように経路の選択を行うことを言う．インターネットでは，小規模なネットワークが細かい網の目のようになることで世界中をつないでおり，ルータ同士が，接続されることで繋がっている．データパケットを宛先のホストにきちんと届けるためには，各ルータが正しい経路にデータを送信する必要がある．各ルーターは，自分の持つ経路制御表（ルーティングテーブル）を参照にして，データを転送します．ルーティングテーブルが異なっていれば．ルーターはデータを正しい方向へ転送できず，データは目的地まで届かないことになる．[3]

2.6 ルーター

ルーターとは，コンピュータネットワークにおいて、二つ以上の異なるネットワーク間を相互接続する通信機器.
ルーターの基本機能は四つある．[4]

- 接続
  ルータは複数の回線種別に対応していることが多く，そのために多くの機種でインターフェースユニットが交換できるようなっている．

- 転送
  ルーターがIPパケットを受け取ると，その中のIPパケット・ヘッダーの宛先アドレスを読み取る．

- 選別
  ルーターは，受け取ったIPパケットに応じて，QoS（Quality of

service）によって優遇して転送したり，フィルタによって転送せずに破棄するなど，パケットの選別機能を持つ．

- 管理
  相互接続された，他のルーターとの通信によって経路情報を交換し合い，常に経路表を最新の状態に保つために管理を行っている．

2.7 トレース

2.7.1 トレースとは

跡をつける，形跡，証拠，小道，なぞる，たどる，追跡する，などの意味を持つ英単語.
プログラミングの分野では，実行すると不具合が生じるプリグラムのどこに問題があるか突き止めるために，命令の実行を順番にたどっていくことを言う．[3]

2.7.2 行列のトレース

ルータの各ポートにあるｎ次元行列で，コンテンツは行列のいくつかのポイントを占めている．コンテンツはｌ×ｍ×ｎのビットの構成された固有の識別番号を持つ．コンテンツがポートを通過するときコンテンツ固有の痕跡｛T1 T2 T3｝をポートに残しておく．経路制御はこれに基づいて行う．図は三次元にあたる．

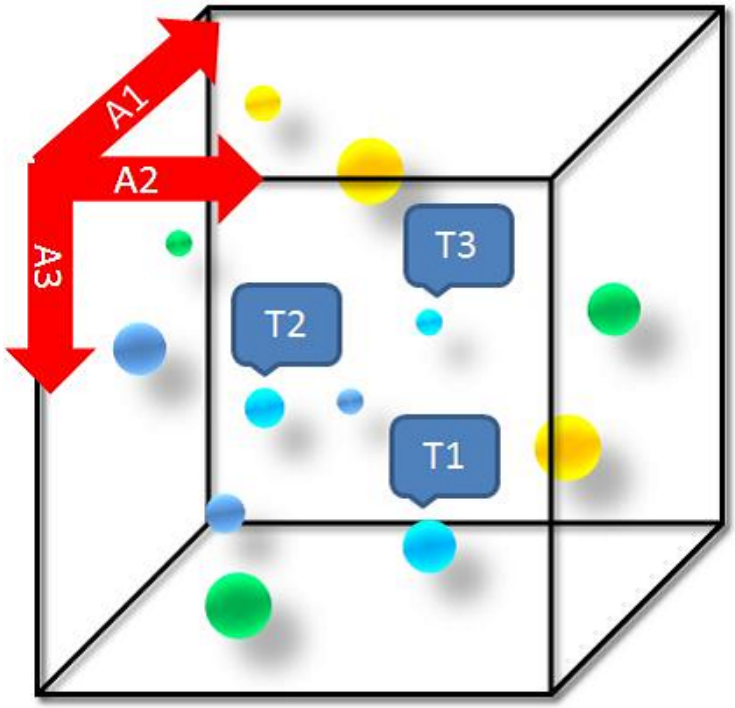

図 2.3: 3 次元行列

2.8 著作権管理に必要な経路制御

経路制御とはネットワーク層やTCP/IPに対応し，相手ホストへの経路を選択する機能をもつルータの最適化を図るものである．

2.8.1 Initial Routing

配布元から消費者へコンテンツの配布，譲渡を行う際に使用される．
流れとして配布元またはユーザーが別のユーザへカプセルを送る．カプセルを送った際にルータは転送する痕跡を行列Mに書き込む．

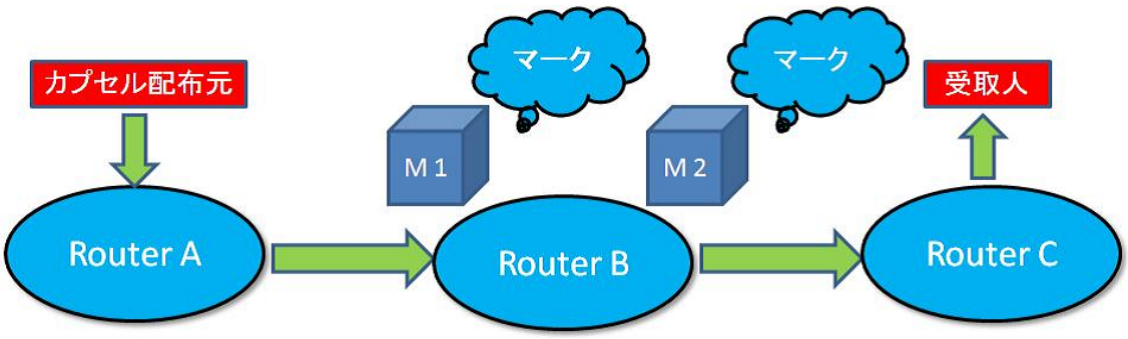

図 2.4: Initial Routing

2.8.2 Forward Routing

配布元から消費者へコンテンツの更新，削除を行う際に使用．流れを図で表示する．ルーターA，ルーターB，ルーターC，ルーターDがあり，ルーターBにはポート1，ポート2，ポート3がある．

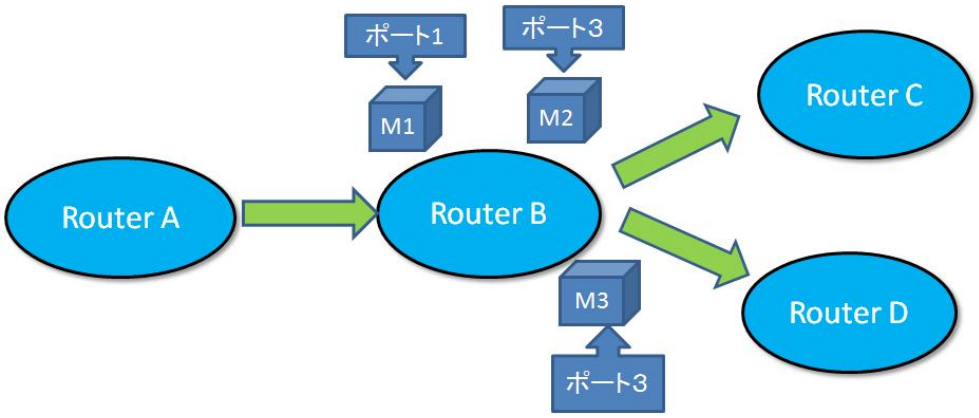

図 2.5: Forwad Routing(step1)

ここでコンテンツが入ってきたと思われるポート１のマークをチェックし，出て行ったと思われるポート２，ポート３のマークも同様にチェックする．

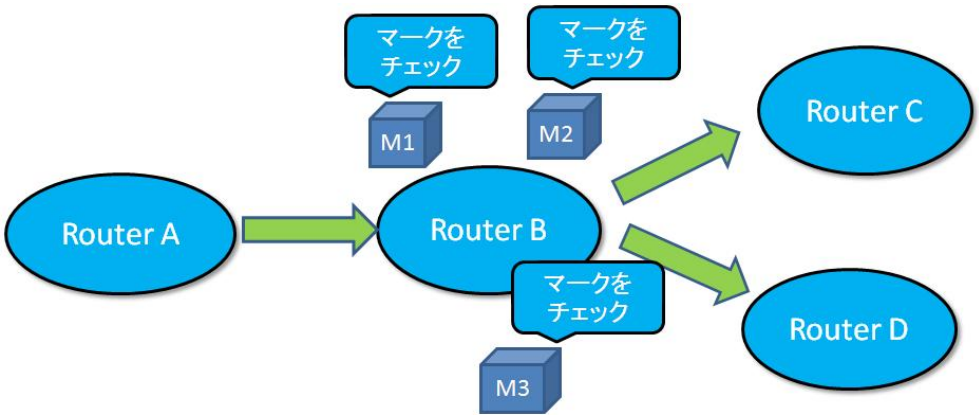

図 2.6: Forwad Routing(step2)

コンテンツの経路を調べるためにチェックしたM1，M2を比較しM1，M3も同様に比較し，M行列内のT1，T2，T3のマークが一致した場合その経路を通過して転送してきたことになる．

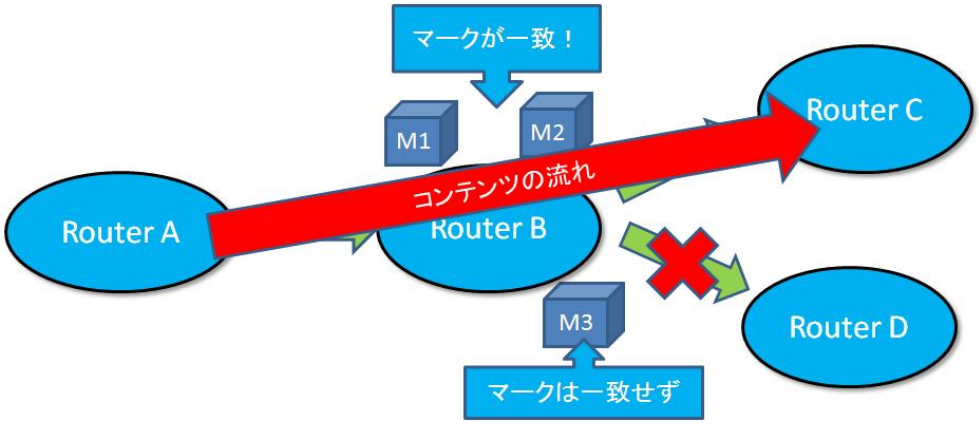

図 2.7: Forwad Routing(step3)

2.8.3 Backward Routing

匿名ルーティングと配給元が通信する時に使用する．ここではInitial Routingの時に残しておいた痕跡をチェックする．ルーターAからルーターCを介してルーターDにメッセージを送ったものとする．

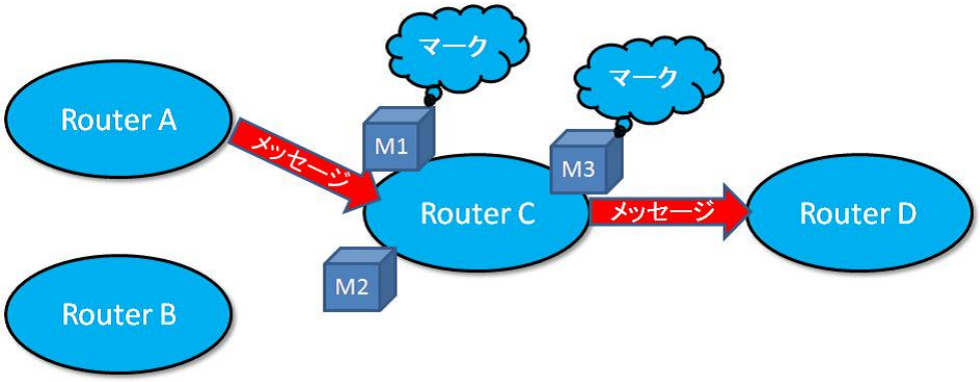

図 2.8: Backwad Routing(step1)

ルーターDの周りにはいくつものルーターが存在しているものとしてこの中から痕跡が一致するポートに接続するルーターへメッセージを送信する．ルーターの各ポートでは痕跡がチェックされる．

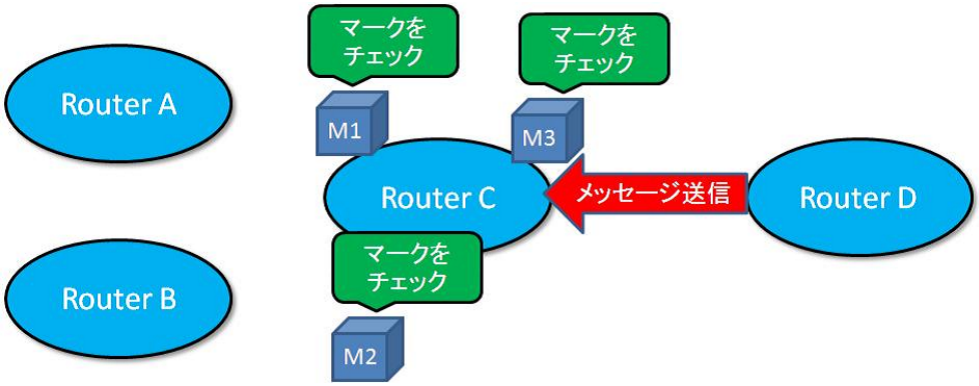

図 2.9: Backwad Routing(step2)

設定した痕跡が存在している場合メッセージがこのポートに転送される.
この場合 M3 と同じマークが存在したのは M1 のためメッセージはルーター A に送信される.

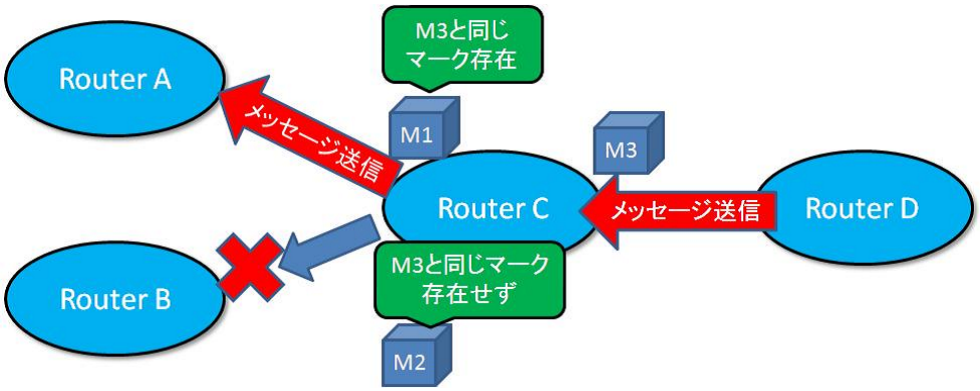

図 2.10: Backwad Routing(step3)

2.9 ポアソン分布

統計学および確率論においてポアソン分布とはシメオン・ポアソンが1838年に確率論とともに発表した．所与の時間間隔で発生する離散的な事象を数える特定の確率変数Nを持つ離散確率分布のこと．時間単位中に平均でλ回発生する事象がちょうどk回（kは0を含む自然数k＝0，1，2…）発生する確率は，次式で表せれる．[3]

$$P(N = k) = \frac{e^{-\lambda}\lambda^{k}}{k!}$$

図 2.11: ポアソン分布式

確率質量関数は以下のように表される．

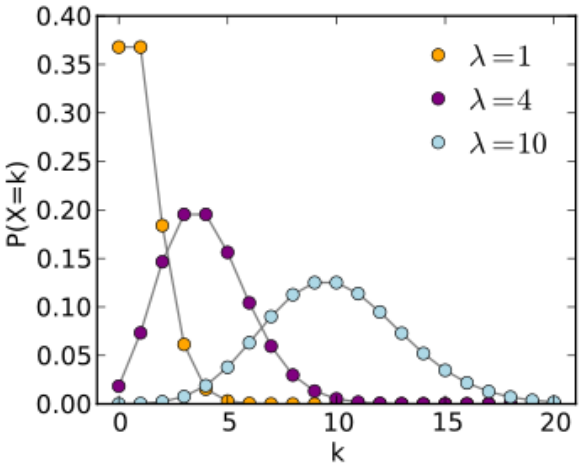

図 2.12: 確率質量関数

第3章　提案

3.1　提案するシステム

映像や音楽ファイルの不正流通が社会問題となっており，恒久的な再生が保障されていないなど消費者の権利に対する不当な制限等改善すべき点が多い．これらを改善するための要素技術のひとつとして匿名通信とその経路制御が必要．しかし，従来のネットワークシミュレーションでは異なる．そこで匿名制御に適したシミュレーションツールを提案する．
前章で挙げた著作権管理に必要な経路制御を使用し，提案するシミュレーションツールを考案する．

3.1.1　シミュレーションについて

- イベント生成モジュール
  どのタイミングでどのノードでどのイベントが発生するか決める．

- ネットワーク生成モジュール
  コンテンツを流通させるための物理的，仮想的なネットワークの構成を決定する．

- シミュレーション駆動モジュール
  シミュレーション全体を制御し，観測結果を収集する．

提案するシミュレーションの構造として図のように提案する.

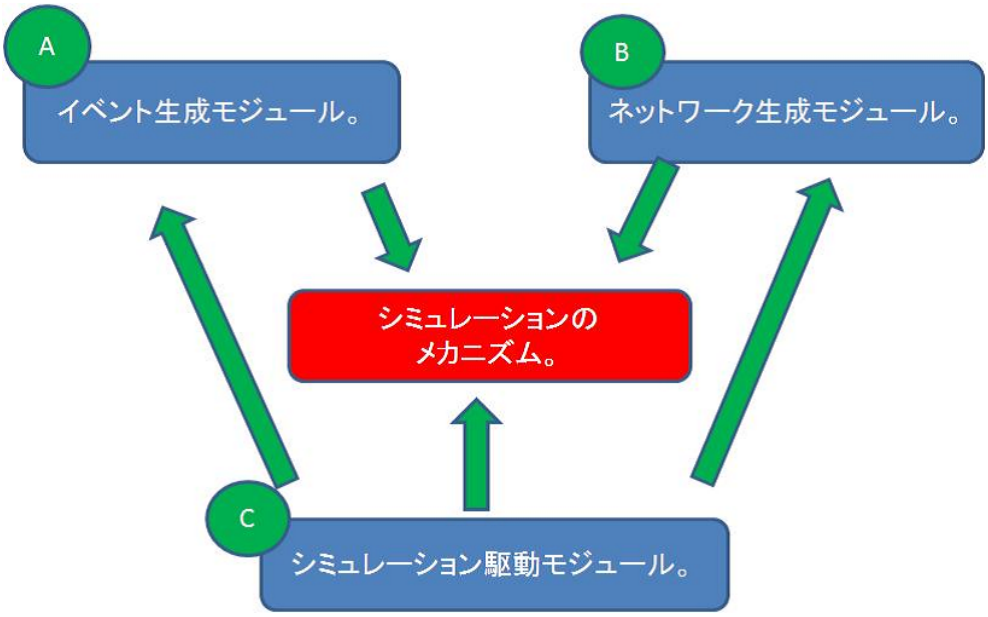

図 3.1: シミュレーターの構造

3.1.2 シミュレーションの流れ

コンテンツの配布時に発生するイベントには確率的な発生を与えるためポアソン分布を使い行う.
発生させるイベントについては以下のもの考える.

- Initial Routing
- Forward Routing
- Backward Routing
- コンテンツ
- 配布・利用条件等の変更
- 二次著作物を配り配布

主な流れとしてはコンテンツの配布が行われた後ポアソン分布を使いイベントを確率的に発生させる．その後は，ノードの状態をイベントが発生したらノードの状態を変化，しなかった場合は変化しないものとする．

3.2 イベント発生時におけるポアソン分布

シミュレーション上のイベントの発生にはポアソン分布を使う．ここではポアソン分布プログラムを紹介する．

```
//イベント発生(ポアソン分布)
int poisson(double lambda)
{
        int k;
        lambda=exp(lambda)*rnd();
        for(k=0;lambda>1.0;k++){
                lambda*=rnd();
        }
        return k;
}

int main(int argc,char **argv)
{
                int i;
                srand((unsigned int)time(NULL));
                for(i=0;i<atoi(argv[1]);i++){
                  printf("%3d %2d\n",i,poisson((double)atof(argv[2])));
                }
}
```

図 3.2: ポアソン分布のプログラム

3.3 提案構造のモジュール

ここでは提案したモジュールについて説明する.

3.3.1 イベント生成モジュール

イベント生成モジュールはシミュレーションを行う際に文字通りイベントを発生させる機能として使用する.
ネットワークが完成し，ノードの位置が決定したらそのノードでイベントを発生させるか決める.

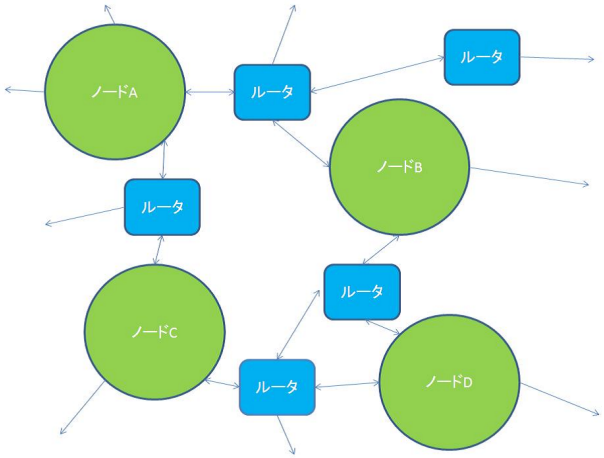

図 3.3: イベントの生成（step1）

イベントの内容を選択し，どこのノードでイベントが発生するか決める．図はInitial Routingのイベントを決めたものである．この場合はノードCにInitial Routingのイベントが選択された．

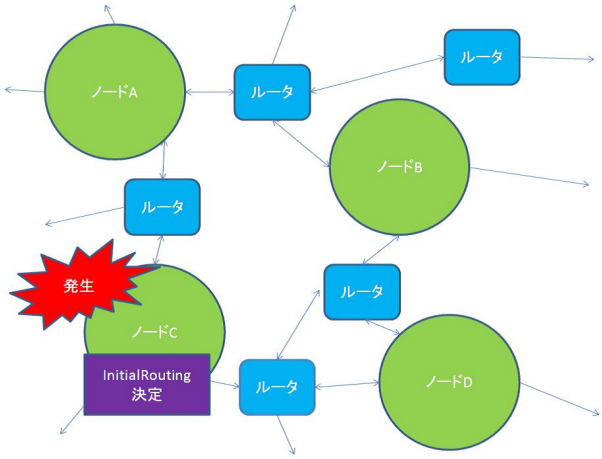

図 3.4: イベントの生成（step2）

同様にして次にForward Routingのイベントの発生を決める．この場合はノードBにForward Routingのイベントが選択された．このようにして順々に発生するイベントを決めていく．

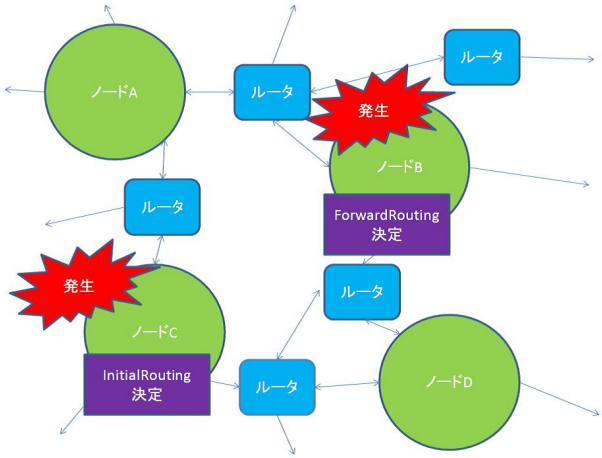

図 3.5: イベントの生成（step3）

3.3.2 ネットワーク生成モジュール

ネットワーク生成モジュールはノードを用いてネットワークを作成．ノードの中に含まれるものとしてルータ，ユーザー，コンテンツ作成者，ユーザーかつ作成者（２次利用）がある．これをもとに枝をつないで構成するが構成する時ランダムに形成される．しかし，形成上に条件を付ける．

- ユーザー同士またはコンテンツの作成者同士等はつなぐことができない．
  ユーザー同士やコンテンツ作成者同士等つないでしまうと匿名性に欠けてしまうため．
- すでに繋がっているものに対しては再度ランダムに行う．
  確率的にいえば既に繋がっている個所に再度当たることがある．その際には重複してしまわないよう再度行う．

以上のことを踏まえ図で説明していく．まずノードと枝の数を指定

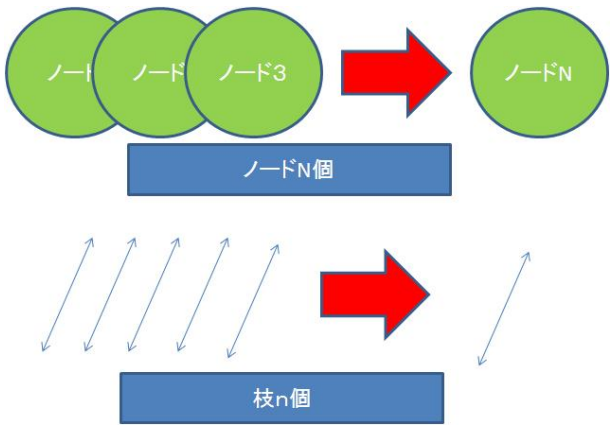

図 3.6: ネットワークの生成（step1）

限られた数の中でネットワークを生成する．条件で挙げたとおりユーザー同士やコンテンツ作成者同士はつなげないものとして考える．図のノードの中は既に決定されているものとして扱う．

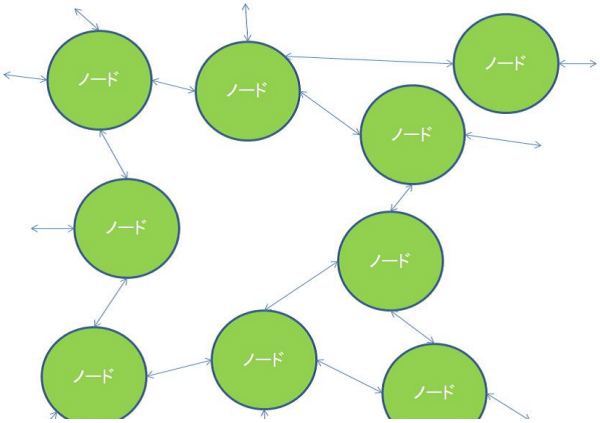

図 3.7: ネットワークの生成（step2）

結果は図のように表示される．

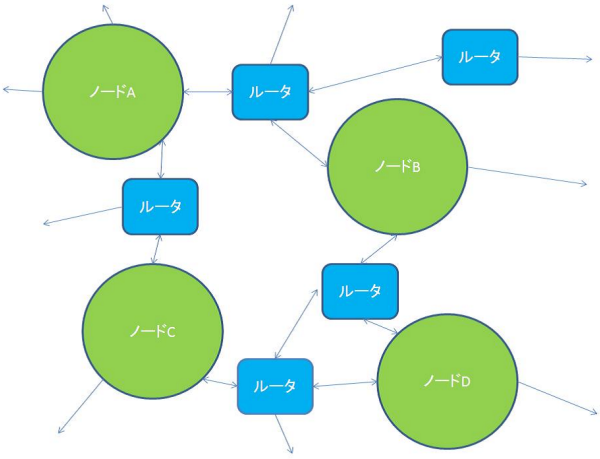

図 3.8: ネットワークの生成（step3）

ルーター以外はノードはどのような形になってもいいのでここではノードのままにしておく．

3.3.3 シミュレーション駆動モジュール

シミュレーション駆動モジュールではここまでのモジュール，シミュレーション全体を制御し，結果を収集する．
イベントの発生のタイミングやネットワークの生成条件をすべてここで指定する．例えば，前セクションで挙げたノードの数や枝の数の指定はここで行っている．イベントの発生も同様で確率的なイベントの発生を指示しているのもこの部分である．

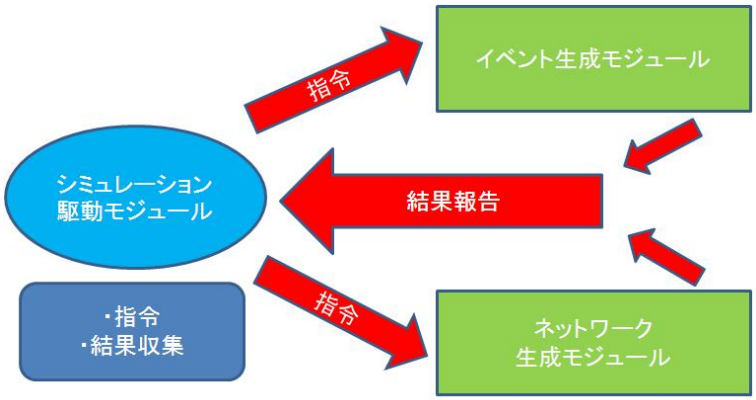

図 3.9: 駆動の仕組み

第4章　まとめ

今回提案したシミュレーションツールは今後は実装を課題とする．実装が可能になることでポートの中に存在する行列にマークされた密度をシミュレーションにより測定し，経路制御のエラー発生率をしく呈することが可能になる．

第5章　謝辞

本研究を行なうにあたり，終始熱心に御指導していただいた木下宏揚教授と鈴木一弘助手，東洋ネットワークシステムズ株式会社の森住哲也氏に心から感謝致します．さらに，公私にわたり良き研究生活を送らせていただいた木下研究室の方々に感謝致します．

2011年2月
原木　章成

関連図書

[1] 五十嵐達治，遠藤直樹，川森雅仁，古原和邦，三瓶徹，中原康治：”ユビキタス時代の著作権管理技術 ”，東京電機大学出版，2006

[2] 山田孔太，木下宏揚，森住哲也，稲積康宏：”エージェントベースの情報カプセルを用いたコンテンツ利用の利便性の向上 ”，SCI2007

[3] ”ウィキペディア（Wikipedia）”
http://ja.wikipedia.org/wiki/

[4] MM-LABO.COM
http://www.mm-labo.cpm/

[5] IT 用語辞典
http://e-word.jp/

[6] ”Microsoft Windows Media デジタル著作権管理（DRM）”
http://www.microsoft.com/japan/windows/windowsmedia/drm/fap.aspx

[7] 安田浩，安原隆一：
”コンテンツ流通の教科書”，株式会社アスキー，2003

[8] ”JASRAC（社団法人日本音楽著作権協会）”
http://www.jasrac.or.jp/profile/copylight/index.html

[9] 酒井雅男，メディア・トゥデイ研究会：
”ディジタル時代の著作権 Q & A ”，ユーリード出版，2003

[10] カプセル化と隠蔽
http://www.nextindex.net/java/capsulate.html

[11] 須田大介：
エージェントによるカプセル化コンテンツの著作権管理，2007年度卒業論文

質疑応答

豊嶋先生から
Q. 今後の実装はどのような言語を使って行うのか.
A. シミュレーションの実装にはC言語を使って行います.