

# レピュテーションを用いた P2P コンテンツフィルタリング性能評価

2005MT018 古田 悠      2005MT128 山下 拓真

指導教員 河野 浩之

## 1 はじめに

P2P ネットワークでは違法コンテンツや個人情報の流出、人気コンテンツやソフトウェアに偽装したウイルスによって感染が広がってしまうなどの問題が起きているため、ピア同士の信頼関係がより重要になってきている。これらの問題は、悪意を持ったプロバイダーからサービスを受けてしまうからであり、各クライアントが優良なプロバイダーと悪意を持ったプロバイダーを上手く区別をすることができないことに原因がある。

本研究では、P2P ネットワークにレピュテーションを用いることで、各プロバイダーにレピュテーションランクという値を持たせる。この値は、プロバイダーの振る舞いに対する信頼を表している。プロバイダーのサービスを受けたクライアントがそのプロバイダーに対してフィードバックすることで決定される。従って、レピュテーションランクはクライアントがプロバイダーを選択する際の判断材料として役立つ。しかし、レピュテーションランクを決定する際に、悪意のあるクライアントによって正しいフィードバックが行われなかったり、レピュテーションランクを悪用してクライアントを騙すプロバイダーが存在する。これは、システムの信用を下げたまま、高く評価されている優良なプロバイダーでさえ信用できない状態になってしまう。そのためにレピュテーションランクを決定するアルゴリズムは、悪意のあるクライアントやプロバイダーが存在する状態であっても、優良なプロバイダーと悪意のあるプロバイダーを区別できるように考えられていなければならない。

我々は、文献 [1] で考慮されている攻撃である Bad-mouthing(常に悪く評価をする)、Ballot-stuffing(常に良い評価をする)に加え Dynamic-user(振る舞いを变化させる)のようなユーザーに対応する既存のアルゴリズムを比較する。その結果からアルゴリズムの性能や適したアルゴリズムを考察する。第 2 章で先行研究内で比較されているアルゴリズムを提示し、その問題点を指摘する。第 3 章では第 2 章で説明した問題に対する解決策になると考えられるアルゴリズムを挙げ、比較するための条件、環境を述べる。そして、第 4 章では第 2 章、3 章で提示したアルゴリズムを比較した結果について述べる。最後に第 5 章でまとめを述べる。

## 2 レピュテーションアルゴリズムにおける先行研究

文献 [1] では、いくつかのレピュテーションアルゴリズムを取り上げ、グリッドのような環境下でサービスの

選択に利用できるように、アルゴリズムを同じ条件で、正確さやオーバーヘッドに関して量的比較を行っている。また、想定されたアルゴリズムへの攻撃に対して適切なアルゴリズムを示し、クライアントがサービスレーティングを故意に誤って報告できるとするならば、これらのアルゴリズムがサービス選択のアプローチになるだろうと述べている。

### 2.1 グリッド環境下の比較に用いたアルゴリズム

以下の表 1 は、先行研究で比較されたアルゴリズムをまとめたものである。Simple Feedback(SF)、Beta Feedback(BF)、Weighted Feedback(WF) は全てのクライアントのフィードバックを扱うのに対して、Beta Filtering Feedback(BFF)、Selective Weighted Feedback(SWF) はフィルタリング機能がある。このフィルタリング機能により不正直なクライアントのフィードバックの影響を少なくする。

表 1 アルゴリズムの機能と特徴

アルゴリズム	フィルタリング	特徴
SF	なし	各クライアントからの評価の総数
BF	なし	ベータ確率密度関数を基盤として計算
WF	なし	評判と直接的な経験を一次結合して計算
BFF	正常範囲外の評価を無視	正常なクライアントリストを作成し計算
SWF	信用できる評価のみを参照	信用値を各クライアントが維持

### 2.2 先行研究における比較条件の問題点

先行研究 [1] では、クライアントによるシステムへの攻撃に対するアルゴリズムが挙げられているが、悪意のあるプロバイダーの振る舞いに関してはシミュレーションが行われていない。

レピュテーションランクを正確に求める為には、クライアントのフィードバックだけでなく、プロバイダーの振る舞いについても考慮し、プロバイダーによるシステムへの攻撃に対応する別のアルゴリズムが必要である。

## 3 P2P 環境下とレピュテーションアルゴリズムを用いた性能比較

### 3.1 Dynamic-user を用いたシミュレーションの提案

先行研究ではプロバイダーが必ず一定のサービスを行うものとしての比較しか行われいない為、良い振る

悪いと悪い振る舞いの両方を行き来するプロバイダー (Dynamic-user) に対応することができない。よって、Dynamic Feedback (DF) を先行研究で紹介されたアルゴリズムに加え、C 言語を用いてシミュレーションすることによって P2P 環境下での性能を比較する。

### 3.2 Dynamic Feedback (DF)

文献 [2] で紹介されているアルゴリズムの特徴は、取引が終了した際に必ずプロバイダー振る舞いが悪くなっていないかをチェックするところにある。

図 1 で示すように、クライアント  $j$  はプロバイダー  $i$  からサービスを受け、フィードバックとしてレピュテーションランクを計算する。また過去に  $i$  のサービスを受け、評価をしているクライアント  $k$  についても評価を与える。その際、評価者の信頼度をプロバイダーのレピュテーションランクに反映させ、それぞれのクライアントの評価に重みをつけている。

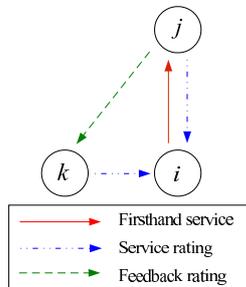


図 1 DF による取引とフィードバックの関係

### 3.3 Dynamic-user の対策

取引が終了した際、必ずプロバイダーの振る舞いの変化をチェックする。そのアルゴリズムは以下の図 2 で実現できる。今までの評判  $SR(i)$  と  $M$  回前までの取引でのランク  $SR_{ref}(i)$  を計算し、比較を行う。 $SR_{ref}(i)$  が低い場合はそのランクを  $SR(i)$  として更新する。

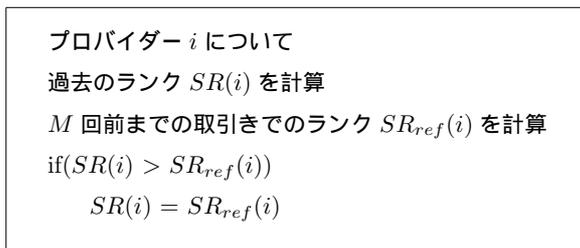


図 2 振る舞いの変化に対応するアルゴリズム

### 3.4 シミュレーションを行う P2P 環境

シミュレーションは図 3 に示すようにクライアントがプロバイダーを選択しフィードバックをする。この取引の繰り返りでレピュテーションランクを求めていく。また、新規のプロバイダーとも取引をする為にクライアントは 10% でランクに関係なくプロバイダーを選択する。それ以外は各アルゴリズムを用い、ランクの高いプ

ロバイダーを選択してフィードバックを行っていく。

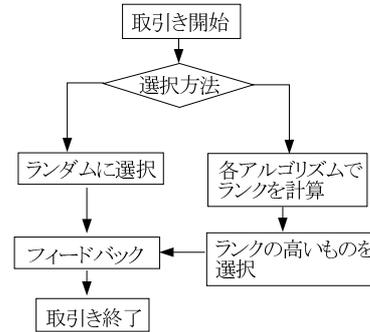


図 3 各ピアの取引の流れ

### 3.5 各シミュレーションにおける比較条件

表 2 では各条件でのシミュレーションの内容を示す。シミュレーション  $SF$  はクライアントが正直なフィードバックをするという仮定で行う。シミュレーション  $BF$  ではクライアント  $k$ 、シミュレーション  $WF$  ではプロバイダーに悪意のあるユーザーが存在するとしてシミュレーションを行う。

表 2 各シミュレーションの比較条件

比較条件	
	ランダムにプロバイダーを選択するものと比較
	ランダムを除いたアルゴリズムのみで比較
	Badmouthing が存在するときの比較
	Ballot-stuffing が存在するときの比較
	Dynamic-user が存在するときの比較

さらに、先行研究よりサービスプロバイダーの数を 40、クライアントの数を 50 とし、クライアントは各サービスプロバイダーに対して 1,000 回リクエストを行うものとする。

## 4 各条件によるシミュレーションの結果

### 4.1 ランダムを含んだシミュレーションの結果

シミュレーション  $SF$  では、ランダムにプロバイダーを選択するもの (Random) と各アルゴリズムを比較する。シミュレーションの結果、悪い振る舞いをするプロバイダーが 70% 存在しても  $SF$ ,  $BF$ ,  $WF$ ,  $SWF$ ,  $BFF$ ,  $DF$  の正答率の差が約 0.02 しかなかった。そのため  $SF$ ,  $BF$ ,  $WF$ ,  $SWF$ ,  $BFF$ ,  $DF$  の中で最も正答率が悪かった  $SF$  と Random のシミュレーション結果を図 4 に示す。

悪い振る舞いをするプロバイダーが 10% 存在する場合には  $SF$  の正答率が Random のものより約 0.1 高い。悪い振る舞いをするプロバイダーの存在する割合が増えるにつれその差は大きくなる。70% のときには、 $SF$  と Random の正答率の差は約 0.6 になる。これは、レピュテーションアルゴリズムによって、正答率の低下を抑えることができるのと同時に、ランダムにプロバイダーを

選択しては、悪意のあるプロバイダーの割合が増えると悪意のあるサービスを受けてしまう可能性が高くなるということがわかる。故に、レピュテーションアルゴリズムが必要であるということがいえる..

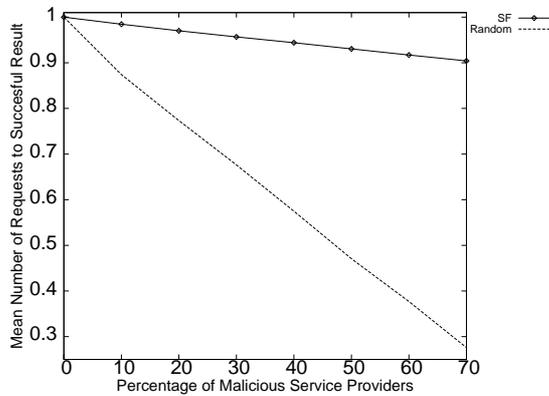


図4 ランダムと各アルゴリズムの比較

#### 4.2 ランダムとSFを除いたシミュレーションの結果

シミュレーション でグラフに示さなかった BF, WF, SWF, BFF, DF の比較を行う。しかし、悪意のあるプロバイダーが 40% 存在するときでも各アルゴリズムに大きな差がないため、50% から 70% までの結果を図 5 に示す。正答率が最も高いのは BF であり、次に BFF, SWF, WF, DF の順となった。しかし BF と DF の正答率の差は悪意のあるプロバイダーが 70% 存在するときで約 0.005 しかなく、大きな差はないといえる。

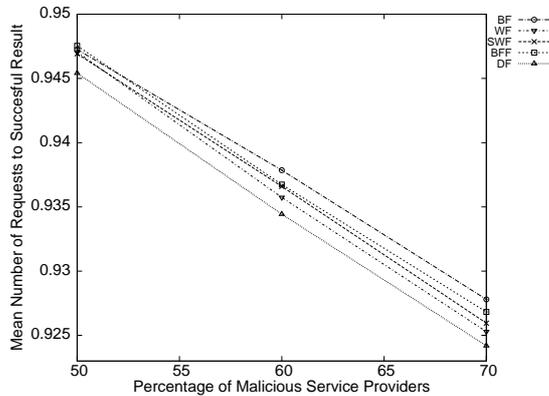


図5 各アルゴリズムの比較

#### 4.3 Badmouthing が存在するシミュレーションの結果

図 6 では、常に悪い評価を下すクライアント Badmouthing が存在する条件下でのシミュレーションを行った結果を表したものである。SF, SWF, は比較的安定しているといえるが, BFF, DF は Badmouthing の割合が増加すると正答率の減少が大きくなってしまい, 70% で正答率が 0.8 を下回る。特に DF に関しては悪い評価に対して過敏に反応することから, 50% で他の

アルゴリズムと約 1.5 の正答率の差が出てしまう。また SWF は、先行研究と同様に信頼できる評価だけを採用しているため、悪意のあるクライアントが増加しても影響を受けにくく、正答率が高い値で推移していると言える。

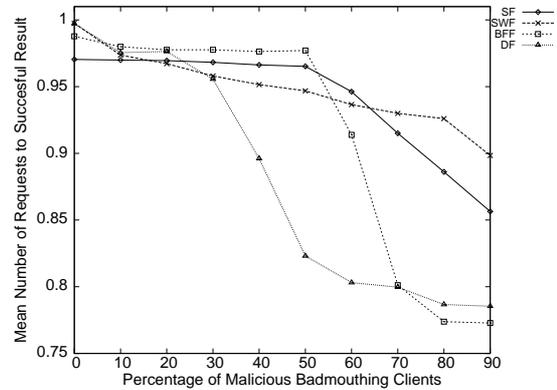


図6 Badmouthing が存在するときの比較

#### 4.4 Ballot-stuffing が存在するシミュレーションの結果

図 7 では、シミュレーション と同様に、SF, BFF, SWF, DF を用いてシミュレーションを行うが、悪意があり常に良い振る舞いをするクライアント Ballot-stuffing が存在する条件下でシミュレーションを行う。本研究では先行研究とは異なる環境でシミュレーションを行っているため、BFF, SWF の正答率が先行研究と異なる値になった。SF, DF は Ballot-stuffing が 80% のときまで正答率を約 0.95 に保っており、高い性能を示した。一方 BFF は 50% までは SF, DF と同じく高い正答率だが、60% 以降は正答率の減少が大きくなっていく。SWF は Ballot-stuffing の増加に伴い徐々に正答率が下がってゆく。

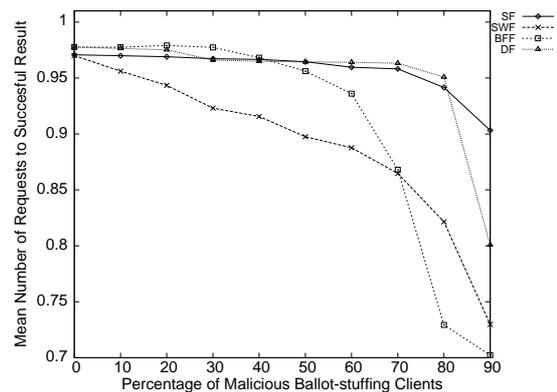


図7 Ballot-stuffing が存在するときの比較

#### 4.5 Dynamic-user が存在するシミュレーションの結果

シミュレーション では、クライアントではなくプロバイダーの振る舞いが増える Dynamic-user に対して

SF, BFF, SWF, DF の正答率を図 8 で示す。DF はもと、プロバイダーの振り舞いに対して対策をしているアルゴリズムであるため、Dynamic-user の存在率が 80% であっても正答率は 0.9 を越えている。しかし、プロバイダーの変化に対して対策のない SF, BFF, SWF は Dynamic-user が増加すると正答率を保つことができない。BFF に関しては、ランダムにプロバイダーを選択したときの正答率と同じ値が得られた。SF, SWF は Dynamic-user が増加すると正答率も徐々に下がっていく。特に SWF はランクを求める際に、扱うフィードバックの数が少ないため正答率の減少の量が他のアルゴリズムに比べて一定にならなかった。

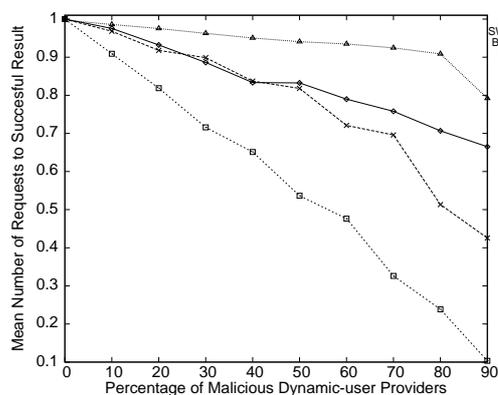


図 8 Dynamic-user が存在するときの比較

#### 4.6 各アルゴリズムの実行時間

各アルゴリズムの実行したときにかかった時間を表 3 に示す。BFF は他のアルゴリズムと比較して明らかに処理に時間がかかるといえる。クライアントの数が 20 の場合で 200 秒の差があるが、クライアントの数が増加するに伴って、その差が次第に大きくなってしまっている。また、実行時間が最も短かったアルゴリズムは SF で、クライアントの数が 20 から 100 に増加しても処理時間は約 0.2 秒しか変わらなかった。これはレピュテーションランクを計算する際に最も単純なアルゴリズムを用いているからだと考えられる。SF, BF, WF, BFF, SWF については先行研究 [1] と同様の値を得ることができた。DF の実行速度は SF には劣るが、フィルタリング機能のあるアルゴリズムと比較すると 20 クライアントの場合で SWF より約 2 秒短い。さらに、クライアントの数が 20 から 100 と増加した場合、SWF の実行時間の差は約 178 秒であるのに対し、DF は 0.4 秒だった。

#### 5 おわりに

本研究では、P2P 環境下においてレピュテーションアルゴリズムを 4 つの条件にわけ、各アルゴリズムを C 言語を用いてシミュレーションを行った。シミュレーションでは先行研究とほぼ同じ結果となった。各アルゴリズムは、ランダムにプロバイダーを選択する

表 3 各アルゴリズムの実行時間

アルゴリズム	実行時間 (sec)		
	20clients	50clients	100clients
SF	0.040	0.090	0.210
BF	0.070	0.160	0.310
WF	0.090	0.230	0.470
BFF	208.61	1408.07	8159.00
SWF	2.290	24.760	180.020
DF	0.040	0.150	0.440

アルゴリズムより高い正答率を得ることができ、レピュテーションアルゴリズムの重要性を示した。また、悪意のあるクライアントが存在しない状態では各アルゴリズムに大きな差はなかった。本研究で追加したアルゴリズムの DF は、Ballot-stuffing または Dynamic-user が存在する条件下において、悪意のあるユーザーが 80% 存在しても正答率は 0.9 を保ち、他のアルゴリズムより優れていた。しかし、Badmouthing に対しては、悪意のあるクライアントが 50% 存在するときの正答率が約 0.8 となり最も低かった。これは、DF の特性である悪い評価に対して過敏に反応することが仇となってしまったといえる。Badmouthing に関しては先行研究と同様に SWF が最も安定した正答率を得た。悪意のあるクライアントが 0% から 90% へ増加しても、正答率の減少は約 0.1 しかなく、他のアルゴリズムに比べ最も少なかった。また、各アルゴリズムの実行時間は、クライアントの数が増えると長くなっていくことが分かった。クライアントの数が 100 のとき BFF の実行時間が約 8,000 秒で最も長く、次に SWF が 180 秒だった。他のアルゴリズムは 1 秒以下で大きな差はなかったが、単純なアルゴリズムの SF が最も短かった。クライアントの数がもっと多くなってくる場合は、SF が優れているといえる。

P2P システムにおいて、クライアントやプロバイダーが単純な動作を行うことは考えにくい。DF のような複雑な環境にも対応できるアルゴリズムが必要になってくると考えられる。今後、さまざまな P2P システムが開発される際に、このシミュレーションはレピュテーションシステム選択の手助けになるであろう。

#### 参考文献

- [1] J. D. Sonnek and J. B. Weissman, "A Quantitative Comparison of Reputation Systems in the Grid," 6th IEEE/ACM International Workshop on Grid Computing, pp.242-255, 2005.
- [2] G. Swamynathan, B. Y. Zhao, K. C. Almeroth, and H. Zheng, "Globally Decoupled Reputations for Large Distributed Networks," Department of Computer Science, June 2007.