Vol.2021-DPS-189 No.6
2021/12/20

オンラインゲームのモデル化によるチート行為の考察

井上 隼[†1]　今泉 貴史[†2]

概要：オンラインゲームはインターネットを介して行うゲームであり，人気のあるコンテンツである．しかし，オンラインゲームにはチート行為と呼ばれる，正規の利用では本来できないことをできるようにする行為が問題になっている．このチート行為の対策は，チート行為が起こることが前提となっており，チート行為をそもそもさせないゲームの作成について，あまり議論されていないという課題がある．
そこで，本研究では，オンラインゲームの抽象化を行い，オンラインゲームのモデルの作成を行う．そして作成したモデルを用いて，チート行為の発生原因についての考察を行う．

1. 序論

オンラインゲームとは，通信を利用して行うゲームであり，メジャーなエンタテイメントコンテンツとなっている．しかし, オンラインゲームにはチート行為と呼ばれる問題がある．チート行為とは, オンラインゲームを構成する要素を, 何らかの方法で不正に利用することにより , 正規の利用では行えないことを出来るようにしてしまう行為を指す．チート行為は, ゲームバランスの崩壊や売り上げの低下につながり, 運営者に損害をもたらし，違法行為として摘発・処罰が必要となるケースもある [1][2][3][4]．そのため運営者は常に対策を行っているが，それらチート行為が起こることを前提とした事後対策が一般的であり，チート行為をそもそも行えないゲームの作成については，あまり議論されていないという課題がある．

そこで本研究では，オンラインゲームの抽象化を行い，オンラインゲームのモデルの作成を行う．そして作成したモデルを用いてチート行為がが行われるかを議論することにより，チート行為の発生原因についての考察を行うことを，本研究の目的とする．

2. オンラインゲームとチート行為

ここではオンラインゲームならびにそこで発生するチート行為について説明したのち，先行研究について述べる．

2.1 オンラインゲーム

オンラインゲームは「コンピュータネットワークを介して専用のサーバや他のユーザのクライアントマシン (PC やゲーム機など) と接続し, 同じゲーム進行を共有することができるソフトウェアを含むサービスを指す.」と定義される [5]．この定義から，オンラインゲームは，

- 複数のマシン同士が通信を行うという物理的な側面
- 同じゲーム進行を共有するという概念的な側面
- オンラインゲームというサービスを運営するというビジネス的な側面

という 3 つの側面を持つことがわかる．以下，オンラインゲームの物理的な構造と，ゲーム進行を共有する方式について述べる．

2.2 オンラインゲームの物理構造

オンラインゲームの物理構造とは，「実際に通信するマシン同士がどういう関係にあるか」という意味である．物理的な構造として，C/S 型と P2P 型の 2 つが挙げられる．

C/S 型は図 1 に表すような構造をしており，データセンターに専用サーバを運営側が所有し，そのサーバを経由してゲームプログラム間のデータの送受信をさせる．C/S 型はサーバが存在しているため，大容量のデータを蓄積することができることが特徴である．

対して，P2P 型は図 2 に表すような構造をしており，端末間で直接ゲームプログラム間のデータを送受信させる．P2P 型はサーバを介さないで通信を行うため，反応速度が C/S 型よりも優れているのが特徴である．

2.3 ゲーム進行の共有方式

先述の通りゲーム進行の共有は, オンラインゲームの必

[†1] 現在，千葉大学大学院融合理工学府
Presently with Graduate School of Science and Engineering, Chiba Uniersity
[†2] 現在，千葉大学統合情報センター
Presently with Institute of Management and Information Technologies, Chiba University

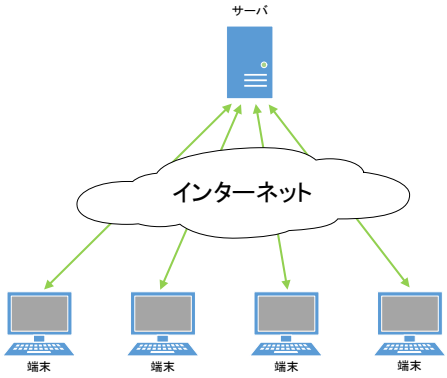

図 1　C/S 型

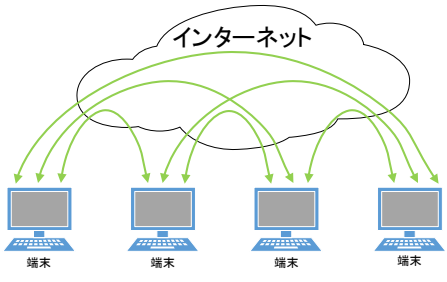

図 2　P2P 型

須事項であり，共有する方法として，「同期型」と「非同期型」の 2 つが存在する．

同期型は，ゲーム進行の状態を，各端末間で一致させ，整合性を完全に取る方式である．同期型は整合性を完全に取らなければならず，大人数でのゲームには向かないが，プログラムの内容を簡単にすることができる．

一方，非同期型は，ゲーム進行の状態が各端末ごとに異なっていることを認め，部分的に整合性を取る方式である．非同期型は整合性を取るのが部分的で済み多くの同時プレイ数に対応できる。一方で，ゲーム進行の状態が各端末ごとに異なっていてもゲームに問題が発生しないように配慮する必要性があり，整合性を取る部分の取捨選択が必要となり，プログラムが複雑化するという問題点がある．[6]

2.4　チート行為

チート行為は前述の通り, オンラインゲームを構成するシステムを, 何らかの方法で不正に利用することで, 正規の利用では本来出来ないことを出来るようにしてしまう行為である．オンラインゲームにおけるチート行為は他のプレーヤーやオンラインゲームの運営者に対して損害を与える行為である．例えば, 対戦型のゲームでチート行為を行い, ステータスを異常に強くしたプレーヤーと対戦したプレーヤーは不快な気持ちにさせる．そしてその不快さからプレーヤーがそのゲームから離れれば，プレーヤーの減少による売り上げの減少が起こりうる．

別の視点から見ると，オンラインゲームの性質上，チート行為は一度手順が確立されれば，瞬間的に手順が拡がってしまう．例えば，もし有料アイテムが不正に大量に所得する手順が拡がれば，その有料アイテムの価値は暴落し，別のプレーヤーに有料アイテムを買ってもらう機会を失うことになってしまう．

2.4.1　ラグスイッチ

ラグスイッチは、通信に一時的に遅延を導入する機器のことである [7].

このラグスイッチを用いて，意図的にオンラインゲームで行う通信を遅延させることで，他プレイヤーの反応を阻害する行為が存在する [8]．2.1 で述べた通り，通信はオンラインゲームをする上で必須である．ゆえに通信を意図的に遅延する行為がオンラインゲーム上で認められるとは考えづらく，チート行為に当たると言える．

2.5　先行研究

チート行為が行われる原因について考察する研究として，小澤や齋藤による研究 [9][10] が挙げられる．これらの研究では，データや機能の配置，データの秘匿性，ゲームの性質をもとに，チート行為の発生原因についての考察を行っていた．

しかしこれらの研究は，オンラインゲームの通信に関す

る議論をしていないため，扱いきれていないチート行為が存在するという課題が存在する．

3. オンラインゲームモデル作成とチート行為の定義

ここではオンラインゲームモデルの作成を行ったのち，本研究で考察の対象とするチート行為について述べる．

3.1 オンラインゲームのモデル作成

3.1.1 考察対象とするオンラインゲーム

本研究では非同期型のオンラインゲームで起きうるチート行為について考察を行う．非同期型のオンラインゲームは，2.3 で述べた通り，設計が複雑化しやすいという問題点がある．ゆえに，非同期型のオンラインゲームにおけるチート行為の発生原因について考察できれば，複雑でわかりづらい非同期型オンラインゲームにおけるチート行為の発生原因を明瞭化でき，チート行為をさせないオンラインゲームの作成に貢献できると考えられる．

なお各端末間で整合性を取るための演算を行う端末は 1 台とする．C/S 型ではサーバ、P2P 型では特定の 1 端末がこれに該当する．

3.1.2 オンラインゲームの定義

前項のようなオンラインゲームを抽象的な形で記述するため，状態遷移機械を用いてプレイヤーとゲームの挙動を定義する．なお，任意のプレイヤ n が操作する端末を”プレイヤ端末 n”，整合性を取るための演算を行う端末を”調停役端末”と呼ぶことにする．C/S 型ではサーバ、P2P 型では特定のプレイヤ端末が調停役端末となり，P2P 型における調停役端末をホストと呼ぶことにする．

プレイヤ数が n のとき，オンラインゲームは以下の要素からなる．

- 入力：$e_{1,0,0}, e_{1,0,1}, ..., e_{n,t,t'_n}$
- 状態：$s_{0,0,0}, s_{0,0,1}, ..., s_{n,t,t'_n}$
- 状態遷移関数：$\delta_0, \delta_1, \delta_2, ..., \delta_n, \lambda$

調停役端末側の状態遷移機械の進行は次のようにする．

(1) プレイヤ端末から入力 e を受信する．
(2) 現在の状態 s_{0,t,t'_0} 及び状態遷移関数 δ_0 を用いて状態 $s_{0,t,t'_0+1} = \delta_0(s_{0,t,t'_0}, e)$ を計算する．
(3) しばらく 1 と 2 を繰り返す．
(4) しばらくしたら，各プレイヤ端末に状態 $s_{0,t+1,0} = \lambda(s_{0,t,t'_{0max}})$ を送信する．
(5) 1 へ戻る．

プレイヤ端末 n 側の状態遷移機械の進行は次のようにする．

(1) プレイヤ n の操作を受け付け，そこから入力 e_{n,t,t'_n} を生成する．
(2) 現在の状態 s_{n,t,t'_n} 及び状態遷移関数 δ_n を用いて状態 $s_{n,t,t'_n+1} = \delta_n(s_{n,t,t'_n}, e_{n,t,t'_n})$ を計算する．
(3) 状態 $s_{n,t+1,t'_n}$ から出力を生成する
(4) e_{n,t,t'_n} を調停役端末に送信する．
(5) 状態 $s_{0,t+1,0}$ を受信するまで 1〜4 を繰り返す．
(6) 状態 $s_{0,t+1,0}$ を受信したら $s_{n,t+1,0} = s_{0,t+1,0}$ を計算する．
(7) 状態 $s_{n,t+1,0}$ を出力する．
(8) 1 へ戻る．

プレイヤ n 側の状態遷移機械の進行は次のようにする．

(1) プレイヤ端末 n の出力を認知する．
(2) 認知をもとに，次に取る行動を判断する．
(3) 判断をもとに，プレイヤ端末 n を操作する．
(4) 1 へ戻る．

本研究では，ゲームに参加するプレイヤが上記のプレイヤ側の状態遷移機械を，各プレイヤが操作する端末が上記のプレイヤ端末側の状態遷移機械を，整合性を取るための演算を行う端末が上記の調停役端末側の状態遷移機械を受け持ち，全体としてのライフサイクルを進行させるようなコンテンツを「非同期型のオンラインゲーム」と定義する．

今回，状態遷移機械によるゲームの定義にあたり，西村による研究 [11] と小澤による研究 [9] を参考とした．

3.1.3 モデルの諸前提

非同期型は，ゲームをプレイする上で問題のない範囲でゲーム進行の状態が各端末ごとに異なっていること許容し，部分的に整合性を取る方式である．

本モデルでは，調停役端末が各プレイヤ端末に状態 $s_{0,t,0}$ を送信し，各プレイヤ端末がこれを受信して $s_{n,t,0} = s_{0,t,0}$ を計算している．これにより，$s_{0,t,0} = s_{1,t,0} = ... = s_{n,t,0}$ となり，すべての端末で整合性を取ることが可能になる．

また，$s_{0,t,0} = s_{1,t,0} = ... = s_{n,t,0}$ で必要な整合性は取られたものとし，そこから各端末において入力 e と状態遷移関数 δ によって状態に差異が生まれても，それはゲームをプレイする上で問題がなく，許容できるものとする．つまり，$0 \leq i, j \leq n$ としたとき，s_{i,t,t'_i} と $s_{j,t+1,t'_j}$ の差異は無視してよく，s_{i,t,t'_i} と $s_{j,t+1,t'_j}$ の差異は無視できないものだとする．

3.2 チート行為の定義

3.2.1 考察対象とするチート行為

本研究では，通信遅延を用いたチート行為について考察する．

通信遅延を用いたチート行為は 2.4.1 で述べた通り，ラグスイッチを用いれば実現可能であるため，手段が確立してあるといえる．ゆえに，通信遅延を用いたチート行為について考察することは，オンラインゲームにおけるチート行為を防止する上で重要だと考えられる．

3.2.2 モデルにおけるチート行為

本研究におけるチート行為は，「ゲームに参加するにあたり，プレイヤに規定されている行動から外れた行動を取

ること」定義する．3.1 で定めた通り，プレイヤに規定されている行動は，

- 出力の認知
- 行動の判断
- 端末の操作

の 3 つである．通信を遅延させることはここに規定されていない行動なので，チート行為ということになる．

続いて，どのような状態においてチート行為が行われるかについて考える．プレイヤがチート行為を行う目的は，ゲームの進行において自身が有利になるか，他プレイヤを不利にするためである．特に通信遅延を用いたチート行為は 2.4.1 で述べた通り，他プレイヤの反応を阻害することで他プレイヤを不利にすることを目的としている．当モデルにおけるプレイヤの「反応」は，出力の変化を認知し，そこから行動を決定し，端末を操作するという一連の動作だと言える．つまりこの一連の動作のうちいずれかが阻害されれば，反応を阻害されたこととなり不利になるということになる．

そこで本研究では，チートを行わなかった際のライフサイクルと行った際のライフサイクルを比較し，チート行為により，どのプレイヤが反応を阻害され不利になったかを考える．そして，チート行為を行うことで，チート行為を行ったプレイヤ以外が不利になる場合，チート行為が行われるとする．

4. モデルとチート行為の考察

ここでは，前の章で作成したモデルを用いて，同じく前の章で定義したチート行為が行われるかをもとに，チート行為の発生原因について考察を行う．

4.1 C/S 型

ここでは C/S 型におけるチート行為について考察する．C/S 型は，図 3 のように表せる．

図 3 で示した C/S 型において調停役端末はサーバであるため，このモデルにおけるライフサイクルは以下のようになる．

(1) プレイヤ i, j はプレイヤ端末 i, j をそれぞれ操作し，プレイヤ端末 i, j はそこから入力 e_{i,t,t'_i}，e_{j,t,t'_j} をそれぞれ生成する．
(2) プレイヤ端末 i, j は入力 e_{i,t,t'_i}，e_{j,t,t'_j} から状態 s_{i,t,t'_i}，s_{j,t,t'_j} をそれぞれ計算，出力する．
(3) プレイヤ i, j は出力を認知し，次の行動を判断する．
(4) プレイヤ端末 i, j はサーバに入力 e_{i,t,t'_i}，e_{j,t,t'_j} をそれぞれ送信する．
(5) サーバは入力 e_{i,t,t'_i}，e_{j,t,t'_j} を受信し，そこから状態 s_{0,t,t'_0} を計算する．
(6) しばらく 1〜5 を繰り返す．
(7) サーバはプレイヤ端末 i, j に状態 $s_{0,t+1,0} = \lambda(s_{0,t,t'_0})$

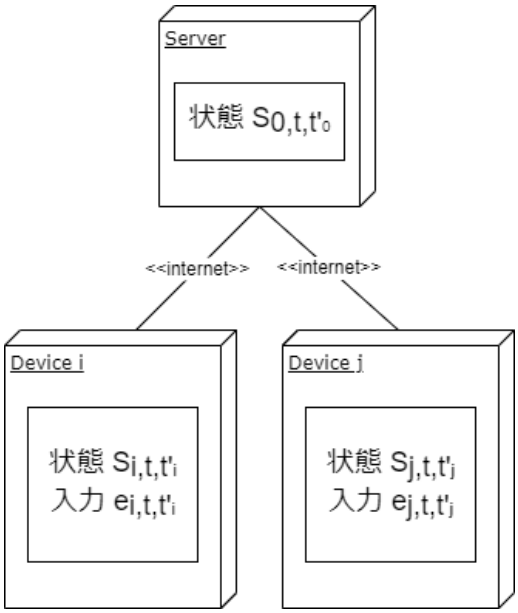

図 3 C/S 型のモデル

を送信する．
(8) プレイヤ端末 i,j は状態 $s_{0,t+1,0}$ を受信し，$s_{i,t+1,0} = s_{0,t+1,0}$，$s_{j,t+1,0} = s_{0,t+1,0}$ をそれぞれ計算，出力する．
(9) プレイヤ i,j はを出力を認知する．
(10)プレイヤ i,j は認知をもとに，次の行動を判断する．
(11)1 へ戻る．

このライフサイクルについて，プレイヤ i が，通信遅延を行うことを考える．

プレイヤ i が通信遅延を行なっている場合，サーバとプレイヤ端末 i はそれぞれの通信を受信できず，ライフサイクルは以下のように変化する．
(1) プレイヤ i,j はプレイヤ端末を操作し，プレイヤ端末 i,j はそこから入力 e_{i,t,t'_i}，e_{j,t,t'_j} をそれぞれ生成する．
(2) プレイヤ端末 i,j は入力 e_{i,t,t'_i}，e_{j,t,t'_j} から状態 s_{i,t,t'_i}，s_{j,t,t'_j} をそれぞれ計算，出力する．
(3) プレイヤ i,j は出力を認知，次の行動を判断する．
(4) プレイヤ端末 i,j はサーバに入力 e_{i,t,t'_i}，e_{j,t,t'_j} をそれぞれ送信する．
(5) サーバは入力 e_{j,t,t'_j} を受信し，そこから状態 s_{0,t,t'_0} を計算する．
(6) しばらく 1〜5 を繰り返す．
(7) サーバはプレイヤ端末 i,j に状態 $s_{0,t+1,0} = \lambda(s_{0,t,t'_0})$ を送信する．
(8) プレイヤ端末 j は状態 $s_{0,t+1,0}$ を受信し，$s_{j,t+1,0} = s_{0,t+1,0}$ を計算，出力する．
(9) プレイヤ j はを出力を認知する．
(10)プレイヤ i,j は認知をもとに，次の行動を判断する．
(11)1 へ戻る．

プレイヤ i が通信遅延を行わなかった場合のライフサイクルと行った場合のライフサイクルを比較する．

通信遅延を行わなかった場合，プレイヤ i は，ライフサイクルの 3 で状態 s_{i,t,t'_i} の出力を，9 で状態 $s_{i,t+1,0}$ の出力を認知する．対して通信遅延を行った場合，プレイヤ i は，ライフサイクルの 3 で状態 s_{i,t,t'_i} の出力を認知する．すなわち，プレイヤ i は通信遅延を行うと，状態 s_{i,t,t'_i} の出力は認知できるが，状態 $s_{i,t+1,0}$ の出力を認知することができなくなる．3.1.3 で述べた通り，状態 s_{i,t,t'_i} と状態 $s_{i,t+1,0}$ の差異は無視できないものである．ゆえに，プレイヤ i 状態 s_{i,t,t'_i} から状態 $s_{i,t+1,0}$ という出力の無視できない変化を認識できなくなることになる．よってプレイヤ i は反応を阻害されたことになり，不利になったといえる．

一方，プレイヤ j は，プレイヤ i が通信の遅延を行わなかった場合と行った場合の双方において，ライフサイクルの 3 で状態 s_{j,t,t'_j} の出力を，9 で状態 $s_{j,t+1,0}$ の出力を認知する．よってプレイヤ j は反応を阻害されず，不利にはならない．

まとめると，プレイヤ i が通信遅延が発生させている間，プレイヤ i は不利になるが，プレイヤ j は特に影響がない．つまりチートを行ったプレイヤのみが不利になるので，チート行為は行われない．

以上から，C/S 型において，通信遅延を用いたチート行為は行われないと言える．

4.2 P2P 型

ここでは P2P 型におけるチート行為について考察する．

P2P 型は，図 4 のように表せる．

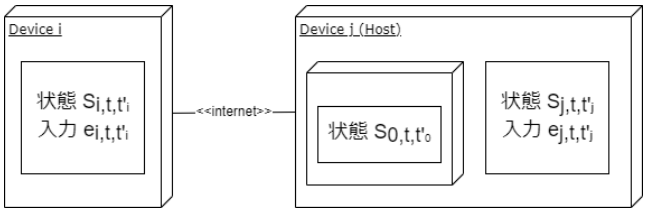

図 4 P2P 型のモデル

図 4 で表した P2P 型において調停役端末はプレイヤ端末 j であるため，このモデルにおけるライフサイクルは以下のようになる．
(1) プレイヤ i,j はプレイヤ端末を操作し，プレイヤ端末 i,j はそこから入力 e_{i,t,t'_i}，e_{j,t,t'_j} をそれぞれ生成する．
(2) プレイヤ端末 i,j は入力 e_{i,t,t'_i}，e_{j,t,t'_j} から状態 s_{i,t,t'_i}，s_{j,t,t'_j} をそれぞれ計算，出力する．
(3) プレイヤ i,j は出力を認知し，次の行動を判断する．
(4) プレイヤ端末 i はプレイヤ端末 j に入力 e_{i,t,t'_i} を送信する．
(5) プレイヤ端末 j は入力 e_{i,t,t'_i} を受信し，それと e_{j,t,t'_j} から状態 s_{0,t,t'_0} を計算する．
(6) しばらく 1〜5 を繰り返す．
(7) プレイヤ端末 j はプレイヤ端末 i に状態 $s_{0,t+1,0} = \lambda(s_{0,t,t'_0})$ を送信する．
(8) プレイヤ端末 i は状態 $s_{0,t+1,0}$ を受信し，プレイヤ端末 i,j は $s_{i,t+1,0} = s_{0,t+1,0}$，$s_{j,t+1,0} = s_{0,t+1,0}$ をそれぞれ計算，出力出力する．
(9) プレイヤ i,j はを出力を認知する．
(10)プレイヤ i,j は認知をもとに，次の行動を判断する．
(11)1 へ戻る．

このモデルにおいて，チートを行うプレイヤは，ホストでない端末とホストの端末のいずれかのプレイヤ端末を操作することになる．そのため，ここではこの 2 つの場合について分けて，チート行為が行われるか考察する．

4.2.1 ホストでない場合

ここでは，チート行為を行うプレイヤが操作するプレイヤ端末がホストでない場合について考察する．

先述の P2P 型におけるライフサイクルについて，図 4 におけるホストでない端末を操作するプレイヤ，すなわちプレイヤ i が，通信遅延を行うことを考える．

なっている場合，プレイヤ端末 i,j 間でそれぞれの通信

を受信できず，ライフサイクルは以下のように変化する．
(1) プレイヤ i,j はプレイヤ端末を操作し，プレイヤ端末 i,j はそこから入力 e_{i,t,t'_i}，e_{j,t,t'_j} をそれぞれ生成する．
(2) プレイヤ端末 i,j は入力 e_{i,t,t'_i}，e_{j,t,t'_j} から状態 s_{i,t,t'_i}，s_{j,t,t'_j} をそれぞれ計算，出力する．
(3) プレイヤ i,j は出力を認知し，次の行動を判断する．
(4) プレイヤ端末 i はプレイヤ端末 j に入力 e_{i,t,t'_i} を送信する．
(5) プレイヤ端末 j は e_{j,t,t'_j} から状態 s_{0,t,t'_0} を計算する．
(6) しばらく 1〜5 を繰り返す．
(7) レイヤ端末 j はプレイヤ端末 i に状態 $s_{0,t+1,0} = \lambda(s_{0,t,t'_0})$ を送信する．
(8) プレイヤ端末 j は $s_{j,t+1,0} = s_{0,t+1,0}$ をそれぞれ計算，出力する．
(9) プレイヤ j はを出力を認知する．
(10)プレイヤ i,j は認知をもとに，次の行動を判断する．
(11)1 へ戻る．

プレイヤ i が通信遅延を行わなかった場合のライフサイクルと行った場合のライフサイクルを比較する．

プレイヤ i が通信遅延を行わなかった場合，プレイヤ i は，ライフサイクルの 3 で状態 s_{i,t,t'_i} の出力を，9 で状態 $s_{i,t+1,0}$ の出力を認知する．対して通信遅延を行った場合，プレイヤ i は，ライフサイクルの 3 で状態 s_{i,t,t'_i} の出力を認知する．すなわち，プレイヤ i は通信遅延を行うと，状態 s_{i,t,t'_i} の出力は認知できるが，状態 $s_{i,t+1,0}$ の出力を認知することができなくなる．3.1.3 で述べた通り，状態 s_{i,t,t'_i} と状態 $s_{i,t+1,0}$ の差異は無視できないものである．ゆえに，プレイヤ i 状態 s_{i,t,t'_i} から状態 $s_{i,t+1,0}$ という出力の無視できない変化を認識できなくなることになる．よってプレイヤ i は反応を阻害されたことになり，不利になったといえる．

一方，プレイヤ j は，プレイヤ i が通信の遅延を行わなかった場合と行った場合の双方において，ライフサイクルの 3 で状態 s_{j,t,t'_j} の出力を，9 で状態 $s_{j,t+1,0}$ の出力を認知する．よってプレイヤ j は反応を阻害されず，不利にはならない．

まとめると，プレイヤ i が通信遅延が発生させている間，プレイヤ i は不利になるが，プレイヤ j は特に影響がない．つまりチートを行ったプレイヤのみが不利になるので，チート行為は行われない．

以上から，P2P 型であり，かつチートを行うプレイヤが操作するプレイヤ端末がホストでない場合，通信遅延を用いたチート行為は行われないと言える．

4.2.2 ホストの場合

ここでは，チート行為を行うプレイヤが操作するプレイヤ端末がホストである場合について考察する．

先述の P2P 型におけるライフサイクルについて，図 4 におけるホストでない端末を操作するプレイヤ，すなわちプレイヤ j が，通信遅延を行うことを考える．

プレイヤ j が通信遅延を行なっている場合,，同様プレイヤ端末 i,j 間でそれぞれの通信を受信できず，ライフサイクルは以下のように変化する．
(1) プレイヤ i,j はプレイヤ端末を操作し，プレイヤ端末 i,j はそこから入力 e_{i,t,t'_i}，e_{j,t,t'_j} をそれぞれ生成する．
(2) プレイヤ端末 i,j は入力 e_{i,t,t'_i}，e_{j,t,t'_j} から状態 s_{i,t,t'_i}，s_{j,t,t'_j} をそれぞれ計算，出力する．
(3) プレイヤ i,j は出力し，次の行動を判断する．
(4) プレイヤ端末 i はプレイヤ端末 j に入力 e_{i,t,t'_i} を送信する．
(5) プレイヤ端末 j は e_{j,t,t'_j} から状態 s_{0,t,t'_0} を計算する．
(6) しばらく 1〜5 を繰り返す．
(7) レイヤ端末 j はプレイヤ端末 i に状態 $s_{0,t+1,0} = \lambda(s_{0,t,t'_0})$ を送信する．
(8) プレイヤ端末 j は $s_{j,t+1,0} = s_{0,t+1,0}$ をそれぞれ計算，出力する．
(9) プレイヤ j はを出力を認知する．
(10)プレイヤ i,j は認知をもとに，次の行動を判断する．
(11)1 へ戻る．

プレイヤ j が通信遅延を行わなかった場合のライフサイクルと行った場合のライフサイクルを比較する．

プレイヤ j は，通信の遅延を行わなかった場合と行った場合の双方において，ライフサイクルの 3 で状態 s_{j,t,t'_j} の出力を，9 で状態 $s_{j,t+1,0}$ の出力を認知する．よってプレイヤ j は反応を阻害されず，不利にはならない．

プレイヤ j が通信遅延を行わなかった場合，プレイヤ i は，ライフサイクルの 3 で状態 s_{i,t,t'_i} の出力を，9 で状態 $s_{i,t+1,0}$ の出力を認知する．対して，通信遅延を行った場合，プレイヤ i は，ライフサイクルの 3 で状態 s_{i,t,t'_i} の出力を認知する．すなわち，プレイヤ j が通信遅延を行うと，プレイヤ i は状態 s_{i,t,t'_i} の出力は認知できるが，状態 $s_{i,t+1,0}$ の出力を認知することができなくなる．3.1.3 で述べた通り，状態 s_{i,t,t'_i} と状態 $s_{i,t+1,0}$ の差異は無視できないものである．ゆえに，プレイヤ i 状態 s_{i,t,t'_i} から状態 $s_{i,t+1,0}$ という出力の無視できない変化を認識できなくなることになる．よってプレイヤ i は反応を阻害されたことになり，不利になったといえる．

まとめると，プレイヤ j が通信遅延が発生させている間，プレイヤ i は不利になるが，プレイヤ j は特に影響がない．つまりチートを行っていないプレイヤのみが不利になるので，チート行為は行われる．

以上から，P2P 型であり，かつチートを行うプレイヤが操作するプレイヤ端末がホストである場合，通信遅延を用いたチート行為チート行為は行われると言える．

4.3 考察のまとめ

この考察によって，P2P 型において，チートを行うプレ

イヤが操作するプレイヤ端末がホストである場合，通信遅延を用いたチート行為が行われることがわかった．一方，オンラインゲームが C/S 型である場合や，P2P 型であってもチートを行うプレイヤが操作するプレイヤ端末がホストでない場合は，通信遅延を用いたチート行為が行われないことがわかった．以上のことから，チートを行うプレイヤが操作するプレイヤ端末に整合性を取るための演算を行わせることが，通信遅延を用いたチート行為の原因になると，考えることができる．

このように，このモデルを用いて議論することで，チート行為の発生原因について考察することが出来た．

5. 結論

本研究では，オンラインゲームの分析と抽象化を行うことで，オンラインゲームのモデルの作成を行った．そして作成したモデルを用いることにより，チート行為の発生原因についての考察を行うことができた．

今後の課題としては，モデルの拡張が挙げられる．本研究では，例えば，複数のプレーヤーが結託してチート行為を行う場合を想定していない．モデルの拡張を行うことで，より多くのチート行為を考察することが期待できる．また，このモデルのチート行為の考察を通し，チート行為が行えないようなオンラインゲームモデルを作成することで，安全なオンラインゲームの作成に貢献したい．

参考文献

[1] 産経 WEST :「モンハン」でチート行為、容疑で大学生を逮捕　摘発は全国初，入手先 ⟨https://www.sankei.com/article/20161004-RVTNMSGVL5MVXBDWPRZIKLML3A/⟩ (2021-11-20).
[2] SankeiBiz :「モンスト」キャラ不正入手で家族養う　兵庫県警が無職男を逮捕，入手先 ⟨https://www.sankeibiz.jp/compliance/news/160229/cpb1602291500002-n1.html⟩ (2011.09.15).
[3] 神奈川新聞 : パズドラのチートツール販売　県警が容疑者ら逮捕，入手先 ⟨https://www.kanaloco.jp/news/social/entry-4126.html⟩ (2011.09.15).
[4] 警視庁 : チート行為はやめましょう，入手先 ⟨http://www.keishicho.metro.tokyo.jp/kurashi/cyber/notes/cheat.html⟩ (2011.09.15).
[5] 中嶋謙互 : オンラインゲームを支える技術-壮大なプレイ空間の舞台裏，技術評論社 (2011).
[6] touge : [CEDEC 2010] ネットゲームの裏で何が起こっているのか。ネットワークエンジニアから見た，ゲームデザインの大原則，入手先 ⟨https://www.4gamer.net/games/105/G010549/20100905002/⟩(2021/11/18).
[7] Bradley Mitchell : Guide to a Network Lag Switch. 2020/11/10，入手先 ⟨https://www.lifewire.com/what-is-a-lag-switch-817481⟩ (2021/11/17).
[8] Vinod Tandon，Jake Devore : DETECTING LAG SWITCH CHEATING IN GAME. ，U.S.Patent(2010).
[9] 小澤 拓海 : オンラインゲームのモデル化とチート行為に関する考察，千葉大学工学部情報画像学科 (2016).
[10] 齋藤貴允 : オンラインゲームの構成によるチート行為の考察，千葉大学大学院融合科学研究科 (2017).
[11] 西村祐介 : P2P ネットワーク上でのオンラインゲームの安全な実行，京都大学大学院情報学研究科修士課程知能情報学専攻 (2006).