Bitswap: Model and Preliminary Analysis

David Grisham

TODO: Ensure periods at the end of all bullet points/lists are consistent

TODO: Figure out cleaner (more maintainable) solution to math mode spacing

---

1 Introduction

TODO: review this section

The InterPlanetary File System (IPFS) is a peer-to-peer data distribution protocol [**TODO: cite**] for sharing hypermedia. IPFS draws inspiration from powerful techniques such as distributed hash tables, sharding, content-addressed data, linked data, and self-certified file systems. At a high level, an IPFS network may be thought of as a Git repository shared in a BitTorrent-like swarm. IPFS has many potential applications, including sharing files within a corporate context, library archival, and, perhaps the most ambitious one, replacing HTTP as the primary file distribution protocol used on the Internet.

Bitswap is the data exchange protocol for the InterPlanetary File System (IPFS). Bitswap's most direct influence is BitTorrent [**TODO: cite**] – like a BitTorrent client, Bitswap determines how to effectively allocate resources (e.g. bandwidth) to peers. In this work, we primarily focus on Bitswap from a game-theoretic perspective. **TODO: finish this off somehow, transition to next section**

TODO: get info in the below paragraph somewhere (if it needs to be)

Our model is meant to reflect this use case of Bitswap as the decision engine implemented by each user in a peer-to-peer distributed file system. In this distributed file system of many users, each user is connect to a set of peers that they trade data with. Every peer has a reputation with every other peer – in other words, for every peer a user has, that user maintains a summary of their interactions with that peer. Then, when deciding how to allocate their resources among their peers at a given time, the user uses these aggregate reputations to provide weights to each of their peers. For example, consider a network of 3 peers, labeled 1, 2, and 3. If peer 2 sends twice as much data to peer 1 as peer 3 sends to peer 1 from time 0 to $t-1$ (and peer 1 sends the same amount of data to both 2 and 3 over that time), then peer 1 might allocate $\frac{2}{3}$ of its bandwidth to peer 2 and $\frac{1}{3}$ to peer 3 at time t.

2 System Model

2.1 Network Graph

We model an IPFS swarm as a network $\mathcal{N}$ of $|\mathcal{N}|$ users. The graphical representation consists of

- *nodes* representing users;
- an *edge* representing a peering between two users; and
- *unweighted*, *undirected* edges.

A user's *neighborhood* is their set of peers, i.e. the set of nodes that the user is connected to by an edge. User i's neighborhood is denoted by $\mathcal{N}_i \subseteq \mathcal{N}$.

TODO: anything else to add here?

2.2 Game Formulation

TODO: may want to update all references 'bandwidth' with 'data'

All users in the network participate in the Bitswap game. The Bitswap game is an *infinitely repeated* (**TODO: complete or incomplete info?**) game where users exchange bandwidth. The game is separated into discrete *rounds* with an individual round denoted by t, where t is a non-negative integer. The game model has the following properties:

- The *players* are the IPFS users in the network $\mathcal{N}$.
- $a_i^t \in \{R, D\}$ is the *action* user i takes in round t, where R and D represent reciprocation and defection, respectively. When a user reciprocates, they allocate resources toward sending data to their peers; when the user defects, they do not send data to their peers.

We also include two simplifying constraints:

1. Each user has the same amount of bandwidth to distribute among their peers in every round.
2. All users always have unique data that all of their peers want. So, when a user allocates bandwidth to a particular peer, that bandwidth can always be fully utilized.

We define b_{ji}^t as the total number of bits sent from user j to peer i from round 0 to $t-1$. Then we can define the *debt ratio* d_{ji} from user j to peer i as

$$d_{ji}^{t} = \frac{b_{ji}^{t-1}}{b_{ij}^{t-1}\ +\ 1}$$

d_{ji}^{t} can be thought of as peer i's reputation in the eyes of user j. This reputation is then considered by user j's *reputation function* $S_j(d_{ji}^{t}, \mathbf{d}_j^{-i,t}) \in \{0, 1\}$, where $\mathbf{d}_j^{-i,t} = (d_{jk}^{t} \mid \forall k \in \mathcal{N}_j, k \neq i)$ is the vector of debt ratios for all of user j's peers in round t *excluding* peer i. The reputation function considers the relative reputation of peer i to the rest of j's peers, and returns a weight for peer i. This weight is used to determine what proportion of j's bandwidth to allocate to peer i in round t. A specific example of a reciprocation function is given in Section 3.

We can now formally define b_{ij}^{t} recursively using the debt ratio and the kronecker delta function δ_{ij}:

$$b_{ij}^{t} = \begin{cases} 0 & t = 0 \\ b_{ij}^{t-1}\ +\ \delta_{a_i^{t-1}R}\ S_i(d_{ij}^{t}\ ,\ \mathbf{d}_i^{-j,t})\ B & \text{otherwise} \end{cases}$$

Considering the recursive case of this definition, we see that the total number of bits sent from i to j increases by $S_i(d_{ij}^{t}, \mathbf{d}_i^{-j,t})B$ (the proportion of i's total bandwidth that i allocates to j in round t) if and only if peer i reciprocated in round $t-1$ (i.e., $a_i^{t-1} = R \implies \delta_{a_i^{t-1}R} = 1$).

Putting all of this together, we define the utility function for player i at time t:

$$u_i^{t} = \sum_{j \in \mathcal{N}_i} \delta_{a_j^{t}R}\ S_j(d_{ji}^{t}\ ,\ \mathbf{d}_j^{-i,t})\ B\ -\ \delta_{a_i^{t}R}\ c$$

where

- $B > 0$ is the amount of bandwidth that a user distributes among its peers; and
- $c > 0$ is the cost incurred when reciprocating (due to bandwidth, energy, etc.).

Putting all of this together, we see that the utility of peer i in round t is the total amount of bandwidth that i is allocated by its neighboring peers, minus the cost to i for providing data to its peers (if it does so). If i reciprocates, then we say that they provide a total of B bandwidth to their neighborhood $\mathcal{N}_i$; otherwise (when i defects), i provides 0 bandwidth in that round.

TODO: fit the following sentences somewhere (or remove it) We start by defining the system in the most general case, then do an analysis on a system subject to simplifying constraints.

TODO: ensure lower bound for t is consistently 0 (and not typo'd as 1)

3 Analysis

For the purposes of this analysis, we make an additional assumption: each user's neighborhood is constant – so any given pair of peers is connected for the entire repeated game. This means that the network topology is static as well.

We now consider consider a specific reciprocation function that user j uses to weight some peer i:

$$S_j(d_{ji}^t\,,\,\mathbf{d}_j^{-i,t}) = \frac{d_{ji}^t}{\sum_{k\in\mathcal{N}_j} d_{jk}^t}$$

We make the additional caveat that $d_{ji}^0 = 1\ \forall\ i,j \in \mathcal{N}$ – think of this as an initial optimistic send for bootstrapping purposes (otherwise, peers would never send each other anything). (**TODO**: best place to say this?)

Plugging this into equation (**TODO**: reference previous u_i^t equation) gives:

$$u_i^t = \sum_{j\in\mathcal{N}_i} \frac{\delta_{a_j^t R}\ d_{ji}^t}{\sum_{k\in\mathcal{N}_j} d_{jk}^t}\ B\ -\ \delta_{a_i^t R}\ c$$

Based on this, we can characterize a single round of the 2-player (symmetric) game with the following payoff matrix:

TODO: subscripts are cut off by table

TODO: decide which of these tables to use (leaning toward first one)

		Player 2	
		D	R
Player 1	D	0	$B(1-\delta_{d_{12}^{t-1}0})$
	R	$-c(1-\delta_{d_{21}^{t-1}0})$	$B(1-\delta_{d_{12}^{t-1}0}) - c(1-\delta_{d_{21}^{t-1}0})$

		Player 2	
		D	R
Player 1	D	$(0,0)$	$(1-\delta_{d_{12}^{t-1}0})(B,-c)$
	R	$(1-\delta_{d_{21}^{t-1}0})(-c,B)$	$(1-\delta_{d_{12}^{t-1}0})(B,-c) + (1-\delta_{d_{21}^{t-1}0})(-c,B)$

When a player defects, they provide no bandwidth and thus their opponent gets a payoff of 0. When a player reciprocates, they base their reciprocation on their opponent's debt ratio. Since there are only two players in this game, the reciprocation function that a peer uses to weight their opponent simplifies to:

$$S_j(d_{ji}^t) = \begin{cases} 1 & \text{if } |\mathcal{N}| = 2 \ \wedge \ d_{ji}^t \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\{i, j\} \in \mathcal{N}$. So, when reciprocating, a player provides B bandwidth to their opponent if and only if their opponent has a non-zero debt ratio (since their opponent will have a weight of 1); if their opponent's debt ratio is 0, they won't provide any bandwidth. This explains the payoffs in the (D, R) and (R, D) cases. The (R, R) case is simply the superposition of the (D, R) and (R, D) cases.

(**TODO**: worth mentioning PD here/should I mention it earlier? what are the differences between this and PD?)

As this is an infinitely repeated game, we want to be able to calculate the discounted average payoff of player i, U_i.

$$U_i = (1 - \epsilon_i) \sum_{t=1}^{\infty} \epsilon_i^{t-1} \ u_i^t(\mathbf{a}^t, a_j^{t-1})$$

where

- $i, j \in \{1, 2\}$ and $i \neq j$.
- $\epsilon_i \in (0, 1)$ is the *discount factor*, which characterizes how much player i cares about their payoffs in future rounds relative to the current round.
- $\mathbf{a}^t = (a_i^t \mid \forall i \in (1, |\mathcal{N}|))$ is the vector of each player's actions in round t.

Further, rather than d_{ij}^t being an aggregate value over all rounds in $[0, t)$, it will only take the immediately preceding round into account. In other words:

$$b_{ij}^t = \delta_{a_i^t R} \ S_i(d_{ij}^t \, , \, \mathbf{d}_i^{-j,t}) \ B$$

Note that although the primary reason for this restriction is to simplify the initial analysis, it also happens to make it more difficult for a peer to free-ride. If b_{ij}^t took into account every round prior to t, then whenenever j plays R they will always provide B bandwidth to i after any round in which i provided any bandwidth to j (**TODO**: wording in this sentence). This is not an issue when b_{ij}^t only considers the previous round, since j effectively forgets i's history beyond a single round. However, when there are more players in the game and each user has more than one peer, competition between peers alleviates this weakness and thus we can extend the lookbehind of b_{ij}^t in larger games without reintroducing the same issue.

We now consider 3 strategies: tit-for-tat, pavlov, and grim-trigger. For each of these, we'll determine whether the strategy is an subgame-perfect Nash equilibrium (SPNE) for the 2-player infinitely repeated game. While it suffices to show a single case of initial conditions that prove a strategy is not an SPNE, all cases will be shown as they may prove useful in understanding the more complex scenarios. (**TODO**: get others' opinions on whether wording in first half of previous sentence is clear) (**TODO**: be sure to follow up on second half of previous sentence later in analysis)

To determine whether a given strategy is an SPNE for the 2-player infinitely repeated (simplified) Bitswap game, we will do the following:

- Consider an initial pair of actions at $t = 0$, (a_1^0, a_2^0).
- Assume player 1 plays the strategy for all rounds.
- Consider two cases:
    1. Player 2 never deviates from the strategy.
    2. Player 2 deviates from the strategy for a single round, at $t = 1$, then plays the strategy in all future rounds.
- Calculate player 2's payoff for the infinitely-repeated game in both cases. If player 2's payoff in case 2 (U_2') is less than or equal to their payoff in case 1 (U_2) for all initial pairs of actions, then the strategy is an SPNE. Mathematically, a strategy is an SPNE if and only if $U_2' \leq U_2$ for all possible initial actions.

An additional note on notation: as the reciprocation function requires information about the game's history to decide how to allocate resources, we will often write is as a function. For example, if player 1's action is $R(0)$, this means that player 1 is playing the reciprocation action (R) but their peer, player 2, provided 0 bandwidth in the previous round – correspondingly, player 1 playing $R(0)$ does *not* incur the cost of reciprocation c for player 1 since they are not providing any resources. Similarly, player 1 taking action $R(B)$ player 2 provided B bandwidth in the previous round (and player 1 will be subject to cost c when reciprocating in this case). When a player reciprocates at $t = 0$, we write their action as $R(B)$ to reflect the fact that they will optimistically send in the initial round (since there would be no history at that point).

3.1 Tit-for-Tat

We start by analyzing the well-studied tit-for-tat (TFT) strategy. A player that uses this strategy always takes the strategy that their peer took in the previous round. So, if player 1 plays action R (D) in round t, then player 2 will play action R (D) in round $t + 1$, and vice-versa.

Let's first consider case where the initial pair of actions is (D, D).

3.1.1 Case 1: (D, D)

Here, both players start by playing the D strategy. We'll first consider the case where player 2 does not deviate from TFT. The strategies at each round then follow:

t	0	1	2	3	4	...
a_1^t	D	D	D	D	D	...
a_2^t	D	D	D	D	D	...

Since neither player deviates from TFT, they both continually play their opponent's previous strategy – the initial state is (D, D), so each player repeatedly plays D in this instance.

We can calculate the payoff of player 2 in this case – notice that, since neither player is ever giving or receiving, they payoff at each round is 0.

$$u_2^t = 0 \; \forall \; t \implies U_2 = 0$$

Now we consider this case where player 2 deviates from TFT for 1 round, at $t = 1$. The resulting action sequence is then:

t	0	1	2	3	4	...
a_1^t	D	D	R(0)	D	R(0)	...
a_2^t	D	R(0)	D	R(0)	D	...

In this case, player 2 deviates from TFT for a single round and plays R at $t = 1$ rather than D, then goes back to playing TFT for all rounds after that. Player 1, on the other hand, never deviates from TFT.

In round $t = 1$, player 2 plays R and bases their reciprocation off of player 1's last action. Since player 1 defected in $t = 0$, they provided 0 bandwidth to player 2. As a result, player 2 provides 0 bandwidth at $t = 1$ since $d_{21}^1 = 0$. Then, at round $t = 2$, player 1 plays R – however, since player 2 provided 0 bandwidth at $t = 1$, $d_{12}^2 = 0$ and player 1 provides 0 bandwidth to player 2 at $t = 2$. Notice that playing $R(0)$ *looks like defecting* to the other player. For example, when player 2 plays $R(0)$ they send nothing to player 1, so as far as player 1 is concerned player 2 might as well have defected.[1]

Each player continues to alternate between D and $R(0)$ forever, all while never providing any bandwidth to their peer. This lets us calculate the payoff of player

[1] Note that it is nonetheless important to have both D and $R(0)$. A player playing $R(0)$ sends nothing *because of* the opponent's behavior in the preceding round, while one playing D sends nothing *regardless of* the opponent's behavior in the previous round.

2 in this second case, which turns out to be the same as the first:

$$u_2^t = 0 \; \forall \; t \implies U_2' = 0$$

Given U_2 and U_2', we can start to discern whether TFT might be an SPNE for this game. We see that $U_2' \leq U_2$ – this means that TFT *might* be an SPNE, but we have to verify that this is the case for all other initial conditions as well.

For the rest of the cases, we simply show the action sequences and the discounted average payoff results

3.1.2 Case 2: (D, R)

When player 2 doesn't deviate from TFT:

t	0	1	2	3	4	...
a_1^t	D	R(B)	D	R(B)	D	...
a_2^t	R(B)	D	R(B)	D	R(B)	...

We can calculate player 2's discounted average payoff in this second case, U_2'. Note that we consider payoffs starting at $t = 1$, not $t = 0$ – this is because we're calculating the payoff that player 2 would perceive if they deviated from TFT for a single round, and that decision is happening at $t = 1$ in our case. (**TODO**: work on the explanation in this last sentence)

Player 2 reciprocates at $t = 0$, which means that they provide B bandwidth to player 1, while player 1 defects. Then, at $t = 1$, player 1 reciprocates and returns the favor by providing B bandwidth to player 2 (who defects in this round). The players alternate between these two states and exchange bandwidth forever. Thus,

$$\begin{aligned} U_2 &= (1 - \epsilon_2)(B - \epsilon_2 c + \epsilon_2^2 B - \epsilon_2^3 c + \ldots) \\ &= (1 - \epsilon_2)(B - \epsilon_2 c)\frac{1}{1 - \epsilon_2^2} \\ &= \frac{B - \epsilon_2 c}{1 + \epsilon_2} \end{aligned}$$

When player 2 does deviate from TFT:

t	0	1	2	3	4	...
a_1^t	D	R(B)	R(0)	R(B)	R(0)	...
a_2^t	R(B)	R(0)	R(B)	R(0)	R(B)	...

Player 1 defects at $t = 0$, so when player 2 reciprocates at $t = 1$ they provide no bandwidth to player 1. However, player 2 provided B bandwidth at $t = 0$, so when player 1 plays R at $t = 1$ they provide B bandwidth in return. Then at $t = 2$, player 2 provides B bandwidth but player 1 provides nothing (since player 2 providing nothing in the previous round). The resulting bandwidth exchanges are the same as in the previous non-deviating case, and thus

$$U_2' = \frac{B - \epsilon_2 c}{1 + \epsilon_2}$$

Thus, $U_2' = U_2$ in this case.

3.1.3 Case 3: (R, D)

When player 2 doesn't deviate from TFT:

t	0	1	2	3	4	...
a_1^t	R(B)	D	R(B)	D	R(B)	...
a_2^t	D	R(B)	D	R(B)	D	...

Thus,

$$\begin{aligned} U_2 &= (1 - \epsilon_2)(-c + \epsilon_2 B - \epsilon_2^2 c + \epsilon_2^3 B - \ldots) \\ &= -\frac{c - \epsilon_2 B}{1 + \epsilon_2} \end{aligned}$$

When player 2 does deviate from TFT:

t	0	1	2	3	4	...
a_1^t	R(B)	D	D	D	D	...
a_2^t	D	D	D	D	D	...

$$\begin{aligned} U_2' &= (1 - \epsilon_2)(0 - \epsilon_2 0 + \epsilon_2^2 0 - \ldots) \\ &= 0 \end{aligned}$$

Comparing these results, we see that $c > \epsilon_2 B \iff U_2' > U_2$. This means that, at the very least, TFT is **not** an SPNE for this game when $c > \epsilon_2 B$.

3.1.4 Case 4: (R, R)

When player 2 doesn't deviate from TFT:

t	0	1	2	3	4	...
a_1^t	R(B)	R(B)	R(B)	R(B)	R(B)	...
a_2^t	R(B)	R(B)	R(B)	R(B)	R(B)	...

Thus,

$$\begin{aligned} U_2 &= (1-\epsilon_2)(B - c + \epsilon_2(B-c) + \epsilon_2^2(B-c) + \ldots) \\ &= B - c \end{aligned}$$

When player 2 does deviate from TFT:

t	0	1	2	3	4	...
a_1^t	R(B)	R(B)	D	R(B)	D	...
a_2^t	R(B)	D	R(B)	D	R(B)	...

Thus,

$$\begin{aligned} U_2' &= (1-\epsilon_2)(B - \epsilon_2 c + \epsilon_2^2 B - \epsilon_2^3 c + \ldots) \\ &= \frac{B - \epsilon_2 c}{1 + \epsilon_2} \end{aligned}$$

Comparing U_2' and U_2, we again get the result that $c > \epsilon_2 B \iff U_2' > U_2$. Thus, **TFT is an SPNE for the two player Bitswap game when $c \leq \epsilon_2 B$.**

3.2 Grim-Trigger

Now we consider the grim-trigger (GT) strategy. A player that uses this strategy plays D in all rounds where their peer has previously played D; otherwise, the player plays R. Formally, for the 2-player game, this strategy is characterized by

$$a_i^t = \begin{cases} D & \text{if } D \in (a_j^0, \ldots, a_j^{t-1}) \\ R & \text{otherwise} \end{cases}$$

where $i, j \in \{1, 2\}$ and $i \neq j$.

3.2.1 Proof that GT is not an SPNE

Consider the initial case of (R, D). The strategy sequences under this initial state with both players playing GT is

t	0	1	2	3	4	...
a_1^t	R(B)	D	D	D	D	...
a_2^t	D	R(B)	D	D	D	...

Thus,

$$U_2 = -(1 - \epsilon_2)\, c$$

When player 2 deviates from GT, the sequence is:

t	0	1	2	3	4	...
a_1^t	R(B)	D	D	D	D	...
a_2^t	D	D	D	D	D	...

Thus,

$$U_2' = 0$$

We see that $U_2' > U_2$. Therefore, **GT is not an SPNE for this game**.

3.3 Pavlov

TODO: is this one worth doing for this case? how do we define the pavlov strategy for this game?

3.3.1 Case 1: (D, D)

3.3.2 Case 2: (D, R)

3.3.3 Case 3: (R, D)

3.3.4 Case 4: (R, R)

4 Discussion

[**TODO: CITE Osborn Intro to Game Theory**] finds that both the TFT and GT strategies are SPNE for the repeated prisoner's dilemma game for a lower bound of the discount factor. We have shown here that neither strategy is an SPNE for the 2-player Bitswap game. The primary difference between this game and the repeated prisoner's dilemma game is that this game is a *trading game*: when a user cooperates, they are providing some sort of resource to their peer. Consider the difference between the (C, C) case in the prisoner's dilemma and the $(R(B), R(B))$ case in the Bitswap game. In the former, each player gets a payoff of 2 for cooperating; in the latter, each player's total payoff is 0 since they both provide (and gain) B bandwidth.

The Bitswap game we have analyzed does not capture some higher-level dynamics that should arise under more general conditions. In particular, having (1) a network with more users and larger peer sets peer user, as well as (2) longer peer-to-peer histories (rather than just single-round lookbehinds), should result in interesting peer dynamics that are not illustrated in this analysis.

4.1 BACKUP

Now we can write d_{ij}^{t+1} in terms of values from round t.

$$d_{ij}^{t+1} = \frac{b_{ij}^{t} \; + \; \delta_{a_i^t R} \; S_i(d_{ij}^{t} \, , \, \mathbf{d}_i^{-j,t}) \; B}{b_{ji}^{t} \; + \; \delta_{a_j^t R} \; S_j(d_{ji}^{t} \, , \, \mathbf{d}_j^{-i,t}) \; B \; + \; 1}$$