A Distributed Autonomous System Design and Implementation: Satoshi Fantasy

Jay Y. Berg

April 2014

1 Introduction

Distributed system designers are notorious for describing their systems first by its known limitations, and then by the practical workaround. These limitations stem from the *Byzantine Generals' Problem* and its impossibility proofs. The generals must all act on the same battle plan which is communicated from general to general through a messenger. Once everyone is in agreement of the plan, they attack. However, there is no way to know for sure if and when everyone is in a agreement. Some of the impossibilities exist due to a possible traitor among the generals, while others are due to time and message latency. [1]

This paper details Satoshi Fantasy, a distributed autonomous fantasy football game named after Satoshi Nakamoto, the creator of bitcoin. Bitcoin is considered the first successful distributed autonomous system. Its main innovation is the the proof-of-work block-chain. Although bitcoin does not solve the the general *Byzantine Generals' Problem* [2]. In the context of a peer-to-peer electronic cash system, it is a"good enough" pragmatic engineered solution to the problems of trust and time. Satoshi Nakamoto has brought esoteric distributed computing design problems into the mainstream consciousness, and with it a newfound public trust for logical, time-tested, distributed consensus protocols.

Overview

Satoshi Fantasy is a game based on fantasy football points, it is designed to be controlled by the most skilled and knowledgeable fantasy football players. A proof-of-skill block-chain is used to keep score with tokens called Fantasybits, which are ultimately tied to a players Fantasy Name. Problems of trust and time, common to all distributed protocols, are solved within this context.

Driving Factors

CJ Spiller, running back for the Buffalo Bills was one of the biggest fantasy busts in 2013. After averaging 6 yards per carry in 2012, he was was being drafted in the first round of most 2013 fantasy drafts [3]. What if a fantasy expert saw this coming? He knew that Spiller was way over-valued. How can he turn this knowledge into value?

Over the past 10 years fantasy football has tripled in size [4]. In 2013, the High Stakes Fantasy Football Players Championship had over 5800 entries and over 6 million in payouts [5]. The game is played by drafting a team, and submitting your lineup each week. Some leagues allow trading players, but the high-stakes public leagues do not. There are also weekly games and a thriving professional expert service industry.

Currently if you wanted to "sell" Spiller, you could *a*) trade him, if you own him; *b*) not draft him; *c*) blog or tweet that he's your top bust predictions; and *d*) not pick him in weekly fantasy games. **Satoshi Fantasy will enable you to earn fantasybits by buying or selling a players end-of-season future points, on an internal decentralized exchanged.** Before being able to trade, you need to acquire some fantasybits.

2 Fantasybits

Fantasybits are created for each fantasy point scored by an NFL player. Unlike digital currency, there is no pre-distribution, distribution schedule, or mining. This feature enables the use of the proof-of-skill block-chain discussed in section 7. Fantasybits are awarded to fantasynames, based on weekly projections of fantasy results.

Projections

During each week of the fantasy season, up until kickoff, projections can be made. This is done by signing a transaction and sending it to the network. Every projection that is in the block-chain on time is eligible for a payout. Once a block containing the consensus results is received, a deterministic distribution algorithm is run to determine the payouts to each player.

Distribution Algorithm: (see appendix B) Let R equal actual results and p_n equal the projections made by each player. Take the difference of the projection from the results $d(p)$. and then get the average difference $\overline{D}$.

$$d(p) = |R - p|$$

$$\overline{D} = \sum_n |R - p_n|$$

Filter out projections that are below average or are 100% or more off the mark $F(d)$.

$$F(d) = \begin{cases} 0 & \text{if } d > \overline{D} \text{ or } d > R \\ 1 & \text{otherwise} \end{cases}$$

Calculate unitpayout X, which will distribute more coins to the better predictions.

$$X = \frac{R}{\sum_n \left((R - d(p_n)) \times F(d(p_n))\right)}$$

Finally, the award function $A(p)$ determines how many fantasy points are awarded for each projection, this is multiplied by 100 for fantasybits.

$$A(p) = X \times (R - d(p)) \times F(d(p))$$

Any points leftover, L, due to bad or no projections get distributed to the block signer .

$$L = R - \sum_n A(p_n)$$

3 State Machine

The underlying protocol changes it behavior based on its current state, there are also different block-chain and transaction rules, see section 6 . A transition to a different state is accomplished with signing and publishing a block. The proof-of-stake verification rules also depend on the transition context. See Figure 1.

Events

A transaction that transfers fantasybits, is only one of many event types. In contrast to bitcoin, the transfer transaction is not a core feature, and its implementation is next to last on he development schedule (see section 10). Following is the list of events, in order of significance.

nameProof sent by new players to claim their fantasyname, it contains proof-of-work data. signed by player

pointProjection sent by players to make projections, it contains the player/week and points. signed by player

dataTransition sent by Data-feed or Skill stake holders, it contains game results, draft results, player meta-data or schedule data. must be signed by a majority of stake holders.

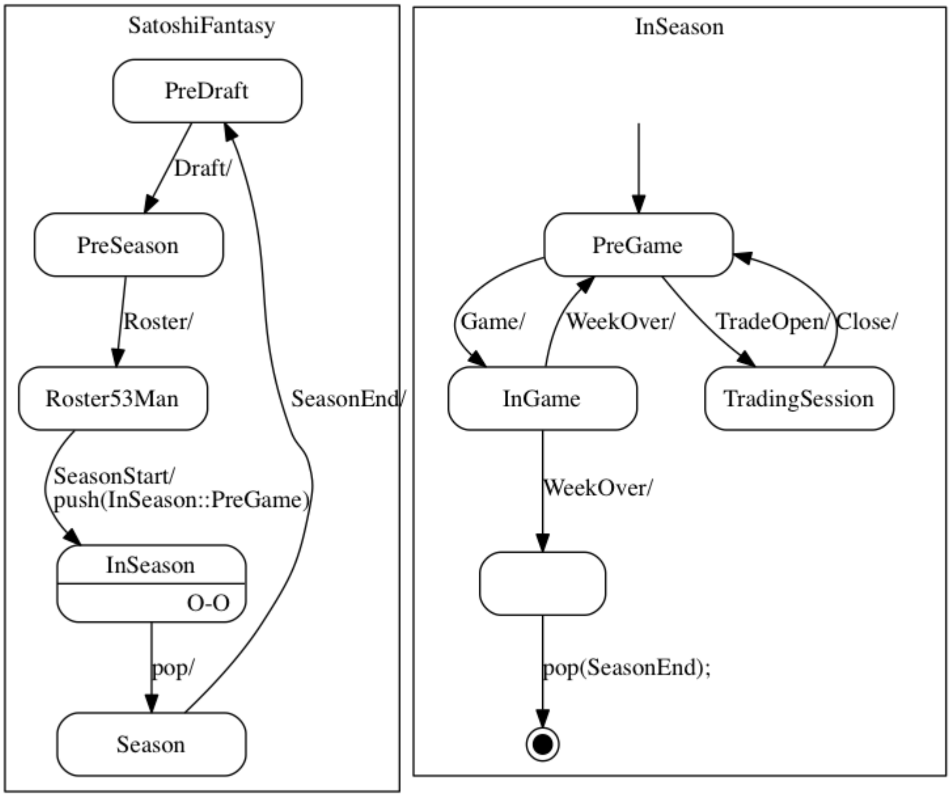

Figure 1: state machine

timeTransition sent by Time-sync or Points stake holders. it contains trading session, exchange events, or any time ordered data. must be signed by a majority of stake holders.

exchangeOrder sent by fantasy players to buy and sell playing players future values. signed by player, sent to designated time-sync engine.

transferTransaction sent by players to player to transfer fantasybits, must be signed by sender.

Deterministic Transactions

A state transition event can trigger multiple events and transactions.

coinbaseTransaction triggered by a dataTransition "WeekOver" event. It will award new fantasybits based on the Distribution Algorithm. See section 2 and figure 1 .

clearingTransaction triggered by a timeTransition "TradeClose" event. It will create transferTransation events, based on the settlement of the sessions trades.

exchangeExecution triggered by timeTransition "TradeOpen" followed by multiple exchnageOrder events, contains fills, and order status.

Satoshi Fantasy In bitcoin, a block chain is an ordered sequence of blocks, with each block containing an unordered set of transactions. Transactions transfer funds between accounts. A block is mined and signed by the minor

4 Fantasy Name

sibyl attack

5 Trading

6 Block Chain

7 Stake Types

8 Transactions

9 Consensus

10 imp

name projection

11 Implementation

References

[1] Nancy Lynch. A hundred impossibility proofs for distributed computing. In *Proceedings of the eighth annual ACM Symposium on Principles of distributed computing*, pages 1–28. ACM, 1989.

[2] kjj. Bitcoin theory (byzantine generals and beyond). https://bitcointalk.org/index.php?topic=99631.

[3] http://www.rotoworld.com/player/nfl/5566/cj-spiller.

[4] JORDAN WEISSMANN. The insane growth of fantasy sports—in 1 graph. http://www.theatlantic.com/business/archive/2013/09/the-insane-growth-of-fantasy-sports-in-1-graph/279532/, sep 2010.

[5] http://www.myffpc.com/ffpccontent/?ffpc-history.

A Scoring Rules

Passing Yards	1 point per 20 yards or .05 points per yard
Passing TD	4 points
Pass interception	-1 points
Rushing Yards	1 point per 10 yards or .1 point per yard
Rushing TD	6 points
Receiving Yards	1 point per 10 yards or .1 point per yard
Receiving TD	6 Points
Reception	1 point per reception
2-point conversion	2 points for passer, rusher, receiver
PAT kick	1 point
Field Goal	3 points for 1-30 yards, .1 point for each additional yard.
Sack	1 point
Takeaway	2 points
Defensive TD	6 points
Safety	5 points
Shutout	12 points
1-6 points allowed	8 points
7-10 points allowed	10 points

B Distribution Algorithm

```
NameValuePairs<double> DistribuePointsAvg::distribute(const Int result) const
{
    double mean = 0;
    vector<Int> diffs;
    diffs.reserve(projections.size());

    for(const auto& pair : projections) {
        Int diff = abs(result-pair.second);
        mean+=diff;
        diffs.emplace_back(diff);
    }

    mean /= projections.size();

    Int maxdiff = min((Int)lround(mean),result);

    Int sum = accumulate(begin(diffs), end(diffs), 0,
        [maxdiff,result](const Int sum,const Int val)
        {
            return sum + ((val < maxdiff) ? result-val : 0);
        });

    double payout = static_cast<double>(result) / sum;
    NameValuePairs<double> award{};

    for (const auto& pair : projections) {
        Int diff = abs(result-pair.second);
        if ( diff < maxdiff )
            award.emplace_back(pair.first,(result-diff)*payout);
    }
    return award;
}
```