Going Beyond Tit-for-Tat: Designing Peer-to-Peer Protocols for the Common Good

Ryan Peterson
Emin Gün Sirer

Cornell University

Introduction

- Current crop of distributed systems are designed for an agent that is

  - selfish
  - operates in isolation with no concept of history
  - has no concept of communal good

State of P2P Practice

- Tit-for-tat is the dominant design paradigm for many distributed systems
  - no client enters a transaction unless it has something to gain
- Easy to implement
  - direct transfer of resources
- Easy to reason about
  - every transaction is mutually beneficial

BitTorrent

- Peers exchange blocks with other peers that have provided them with high bandwidth in the past
- Optimistic unchoking provides discovery of other peers

Samsara

- Backup system where peers store data for each other
- Peers store data on other peers in exchange for claims
- Claims may be traded freely among peers
- Additional mechanisms like spot check enforce compliance

SHARP

- Resource management system where peers exchange CPU cycles
- Every node issues its own tokens to other nodes in exchange for service
- Tokens may be redeemed in the future
  - tokens bind peers because they are redeemable by a specific peer
  - requires coupling with a reputation system

Tit-for-Tat

- Fundamental basis is barter
  - binds the pair of peers that will exchange resources
- Future claims and currency systems loosen the binding
  - exchange might be immediate or delayed
  - claims may be binding or delegated

Problems with Tit-for-Tat (1/2)

- Requires synchronized demand
  - A and B must have resources to offer each other to enter into a transaction
  - in BitTorrent, slow startup phase due to lack of resources to trade
- Delayed exchange systems address this problem, but introduce new ones
  - bankruptcy, debtor might default
  - inflation, claims lose value

Problems with Tit-for-Tat (2/2)

- Valuation is very difficult with nonhomogeneous goods
  - in BitTorrent, seeding in swarm A provides no benefit in swarm B
- Requires discovery
  - in big swarm, difficult to match peers optimally
- Requires policing
  - uncoordinated actions create system-wide vulnerabilities
  - in BitTyrant and BitThief, peers exploit lack of global knowledge

A [illegible]ve

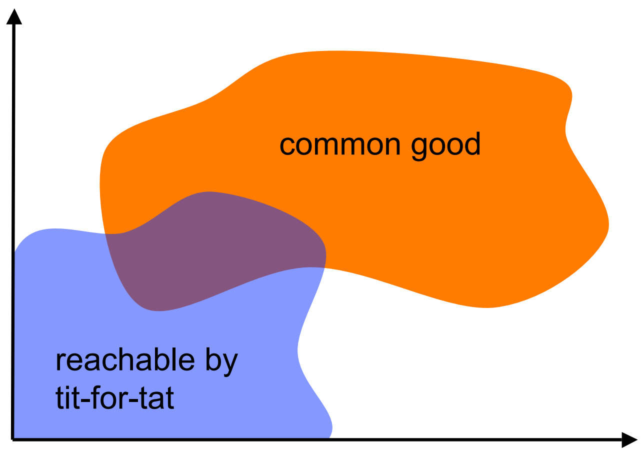
Common Good Paradigm

- Systems where peers act to uphold a global objective function
  - peers may temporarily act against their immediate interests
  - a good global objective function ensures that in the long term every peer receives a benefit

Tit-for-Tat vs. Common Good

- In tit-for-tat, peers are limited to behaviors that are in line with their immediate self-interests
  - some points of operation may be unreachable
- Instead, compute the optimal point of operation and incentivize peers to operate at that point

Operating State Space



Challenges

- Compute the common good
- Enforce behavior to uphold the common good

How?

- Define a system-wide objective function for the common good
  - application-specific
- Aggregate global system state to compute optimal point
  - incentivize peers to report their state
- Police peers
  - ensure peers operate for the common good

AntFarm: Content Distribution for the Common Good

- Content distribution is a critical application
  - accounts for most of Internet bandwidth usage
  - many entities looking to distribute media
- A system for distributing multiple media files
  - swarming downloads similar to BitTorrent
  - optimal use of bandwidth for multiple swarms
- Common good objective function
  - minimize average file download time

A Token-based Solution

- Authority issues peers spend-once tokens
- Peers exchange tokens with other peers in exchange for resources
- Peers are rewarded for sending spent tokens back to the authority
- Authority receives updates from peers!

Conclusions

- Systems where every transaction needs to be mutually beneficial are fundamentally limited
- Designing for the common good can move the system to an optimal point of operation not otherwise reachable