Virtual communities, Peer-to-Peer Networks

Basic introduction to the (sociology and) economics of file-sharing in BitTorrent systems

name: David Hales
date: Jan 30th, 2009
event: Spring School Human Technology Interaction

What has sociology or economics got to do with peer-to-peer systems?

P2P systems *are* socio-economic systems

- Peers cooperate collectively to achieve their goals
- No peer in the system controls everything
- Performance results from interactions
- At the end-of-day users (people) are in control
- Sociology and economics has studied such phenomena - we should steal what we can!

OK but what use is this to me?

Knowing some of the economic background should help you to understand:

- the basic social/economic theory behind P2P like Tribler
- how this informs designs
- how such designs might be improved
- how to assess new developments and designs
- how to evaluate / compare different approaches

It is also a fascinating area in itself:

- If you are interested you can look-up the terms given in *red italics* on Wikipedia for good introductions

Individualism v. Collectivism

In socio-economic systems individual interests may conflict with collective interests:

- e.g. over exploitation of a common resource (a river, a field, the atmosphere etc.)
- e.g. banks - lending (to those who they know can not repay) to gain a commission by selling on the debt to other banks
- e.g. P2P file sharing system - downloading more than uploading

Individualism v. Collectivism

Consider a P2P file sharing system:

- It is in the *collective interest* for all to upload to others so everyone gets the file quickly
- But it is in the *individual interest* to save bandwidth by only downloading and hence free-riding on others
- Free-riding (or free-loading) is a perennial problem in P2P file-sharing systems
- Any efficient system needs to tackle it in some way

The tragedy of the commons

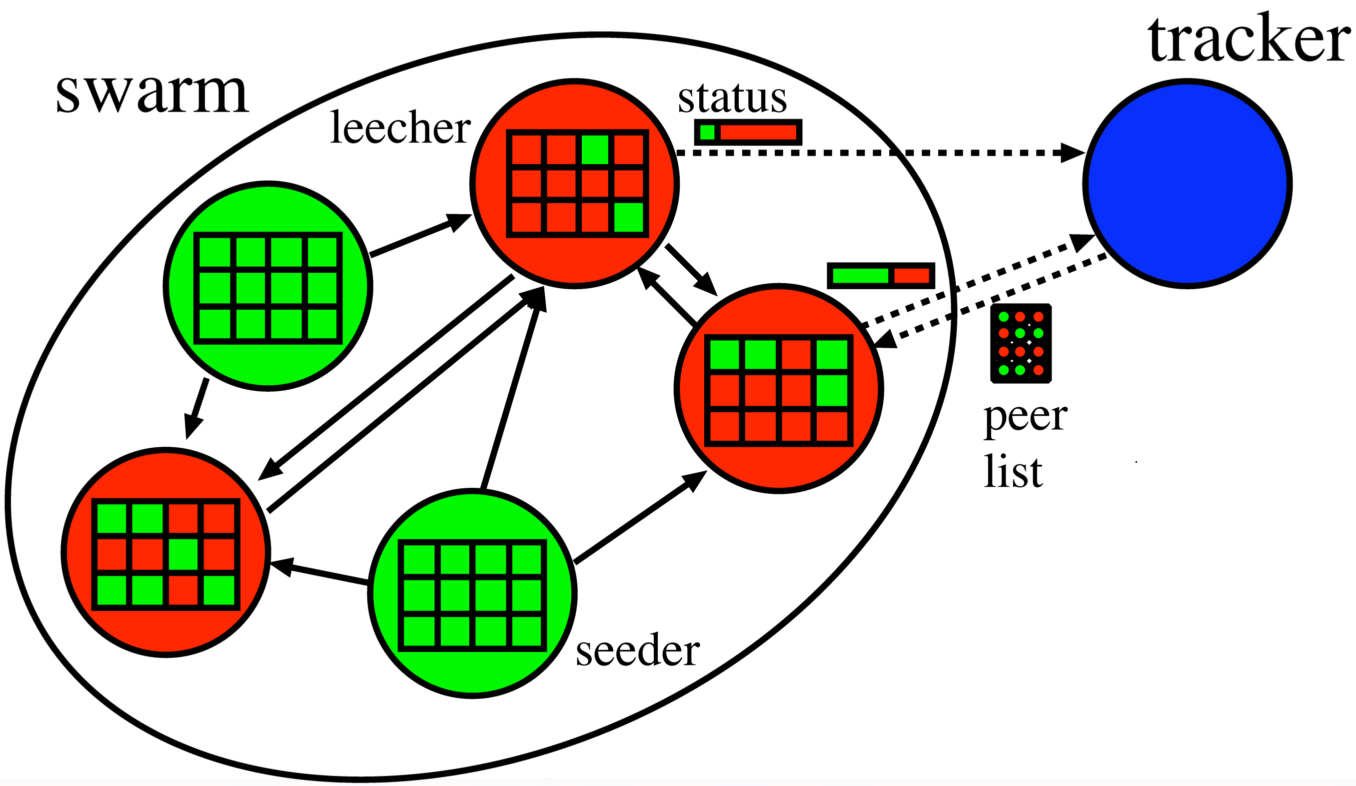
- These kinds of situations have been termed “commons dilemmas” or “common pool resource dilemmas”
- Called “dilemmas” because we would all be better off if we “did the right thing” but there is an individual incentive to do the wrong thing
- G. Hardin (1968) summarized the issue in his famous paper: “The *Tragedy of the Commons*”
- These kinds of situations occur in P2P file-sharing systems like *BitTorrent*

Some BitTorrent Terminology

- Swarm: set of peers interested in a file
  - file is split in smaller chunks called pieces
  - seeder: holds a full copy of the data
  - leecher: holds only a part of the data (initially nothing)
- Tracker: centralized manager
  - keep track of all peers in the swarm
  - return list of current peers in swarm
- Torrent file: meta-data
  - contains pointer to tracker hosting the swarm
  - details about the file - hash, no. of pieces, size etc.

BitTorrent Protocol

- Get a list of other peers in the swarm from the tracker
- Ask peers their list of pieces and tell them what is yours
- Exchange pieces with appropriate peers



How to avoid the commons tragedy?

Central enforcement of correct behaviour

- require centralised agencies and policing
- ability to identify and track individuals centrally
- not suitable for pure P2P (but used with private trackers - see next talk on BarterCast)

Decentralised methods

- self-policing producing incentives for cooperation
- do not require centralised coordination
- more suitable for pure P2P
- can apply ideas from "*game theory*"

What is game theory?

A way to mathematically analyse games assuming we know:

- number of players
- possible moves they can make (strategies)
- outcome of game based on players moves (pay-off)
- desirability of game outcomes for each player (utility)

What game are you playing?

Games can be categorised into two types:

1) Zero-sum games

- when one player wins another loses
- summing the final utilities of players = 0
- e.g. poker, chess, monopoly etc.

2) Non-zero-sum games

- utilities do not always sum to zero
- both players may lose or both may win
- considered to capture social / economic realities
- e.g. tragedy of the commons examples

Capturing a commons tragedy with a simple game

Consider a game composed of two players:

- each player:
  - has choice of one move (C or D)
  - makes a single move then the game ends
  - does not know how the other will move
  - gets a payoff (or utility) based on how they moved and how the other player moved
- for certain payoff values this game can, minimally, capture a form of commons tragedy (or dilemma)
- a classic such game is called the *Prisoner's Dilemma*

The Prisoner’s Dilemma - “payoff matrix”

Game is a PD when: T > R > P > S and 2R > T + S

Player 1 \ Player 2	C	D
C	(3) R / R (3)	(5) T / S (0)
D	(0) S / T (5)	(1) P / P (1)

The Prisoner's Dilemma - example games

Players =>	P1	P2	P1	P2	P1	P2	P1	P2
Moves =>	C	C	C	D	D	C	D	D
Payoffs =>	R	R	S	T	T	S	P	P
Values =>	3	3	0	5	5	0	1	1
Total =>	6		5		5		2	

A contradiction between collective and individual interests

Game theory says defect!

Game theory assumes players are:

- rational - attempt to maximise *their* utility
- selfish - don’t care about the other guy
- knowledgeable - have complete information
- clever - have unlimited computational time

Given these assumptions it can be proved:

- agents will select equilibria where no player will improve by changing strategy unilaterally
- many games have such equilibria - by the famous John Nash (so-called *Nash Equilibrium* - NE)
- the NE for the PD is DD (all defect)

Iterated Prisoner's Dilemma

Previous example "one-shot" PD but:

- real world interactions often repeated
- might meet the guy you just ripped-off in the future
- allows for more complex sequence of strategies based on past interactions with others
- can punish someone tomorrow for defecting against you today - "the shadow of the future"

Iterated PD (IPD) captures this and, as we will see, maps well onto P2P file-sharing protocols like BitTorrent

What is the rational thing to do in the IPD?

Traditional game theory has trouble here:

- cooperative equilibria exist in infinitely repeated games but not in finite games of known length
- many equilibria exist and it is not clear which one would be chosen by rational agents
- In all cases defection on every round is still a equilibrium even when cooperative equilibria exist

For these reasons *Robert Axelrod* (political scientist), in the late 70's, decided to find out what kinds of strategies worked well in the IPD by using computer simulation

Axelrod's Tournament - programs as strategies

Axelrod organised an open IPD tournament:

- Academics were asked to submit programs (BASIC or FORTRAN) that would play the IPD against each other
- Nobody knew competitors code
- The only input would be the on-going past history of the game (a string of C's and D's)
- The aim was to get the highest score (utility) based on round-robin playoffs between all pairs of programs
- Axelrod's aim was to see which programs did best against all the others and understand why
- He wrote-up his results in the famous book "the evolution of cooperation"

Axlerod’s Tournament - what happened?

Basic results were:

- many strategies were submitted (complex and simple)
- the one with the highest overall score turned out to be simple: *tit-for-tat* (TFT) or “look back”
- starts playing C, then “looked back” at the last move made by opponent and copied that move
- submitted by Psychologist Anatol Rapoport
- didn’t “win” against each strategy but did better overall on average against all strategies
- TFT mechanism an example of “*reciprocal altruism*” (Robert Trivers)

What has this got to do with BitTorrent?

In the *BitTorrent protocol*:

- TFT-like method used for sharing files
- nodes form groups interested in a particular file (swarms) and swap or “barter” pieces with each other
- if a node does not upload data then this can be compared to playing defection
- it is punished in the future by being “choked” - not getting upload from others
- even if you hack your client to be selfish the chances are the standard TFT-like protocol will do better overall
- *Bram Cohen* - original BT designer - inspired by Axelrod’s tournaments

The Global Ecology of BitTorrent Clients

Many *bittorrent clients* exist in "the wild"

- Bittorrent 6 (from Bittorrent.com, formally utorrent)
- Others: Azureus, ABC, Transmission, many others...
- Tribler (of course)
- bad guy clients: BitThief, BitTyrant

Hence:

- The current bittorrent ecosystem is a ***global on-going experiment***, like Axelrod's, but with huge user base and rich interactions (not just TFT) incredible strategy sophistication
- This is unprecedented and will surely lead to new economic theory - in general!

BitTorrent Clients

BitTorrent client	FOSS	Linux/Unix	Windows	Mac OS X	IPv6[1]	Programming language	Based on	Interface	Spyware/Adware /Malware-free
ABC	Yes	Partial	Yes	No	buggy[2]	Python	BitTornado	GUI and web	Yes
Acquisition	No	No	No	Yes	?	Objective-C and Cocoa	Limewire	GUI	Yes
Anatomic P2P	Yes	Yes	Yes	Yes	No	Python	BitTornado	GUI and old CLI	Yes
Arctic Torrent	Yes	No	Yes	No	No	C++	libtorrent	GUI	Yes
aria2	Yes	Yes	Yes	Yes	?	C++	-	CLI	Yes
Azureus	Yes	Yes	Yes	Yes	Partial[3]	Java and SWT	-	GUI, CLI, Telnet, Web, XML over HTTP remote control API	Yes
BitComet	No	No	Yes	No	No	C++	?	GUI	Yes [4]
BitFlu	Yes	Yes	No	Yes	Yes	Perl	-	Telnet and Web	Yes
BitLet	Planned	Yes	Yes	Yes	?	Java and JavaScript	-	Web XHTML	Yes
BitLord	No	No	Yes	No	No	C++	BitComet	GUI	Adware
BitPump	No	No	Yes	No	No	C++	-	GUI	Yes
Bits on Wheels	No	No	No	Yes	No	Objective-C and Cocoa	-	GUI	Yes
BitSpirit	No	No	Yes	No	No	C++	BitComet	GUI	Yes
BitThief	No	Yes	Yes	Yes	?	Java	?	GUI	Yes
BitTornado	Yes	Yes	Yes	Yes	Yes	Python	BitTorrent	GUI and CLI	Yes
BitTorrent 5 / Mainline	Yes	Yes	Yes	Old version	No	Python	-	GUI and CLI	Yes
BitTorrent 6	No	No	Yes	No	Yes	C++	µTorrent	GUI and CLI	Yes
BitTyrant	Yes	Yes	Yes	Yes	Partial [3]	Java and SWT	Azureus	GUI, CLI, Telnet, Web, XML over HTTP remote control API	Yes
Blizzard Downloader	No	No	Yes	Yes	?	?	BitTorrent client for early version	GUI	Yes
Blog Torrent	Yes	No	Yes	Yes	?	?	BitTorrent client for early version	GUI	Malware-Status: unknown
BTG	Yes	Yes	Partial[5]	Yes	No	C++	libtorrent	CLI, GUI and web	Yes

Tribler additions to BT incentive mechanisms

Incentives for seeding:

- BT relies on nodes uploading pieces even when they have all pieces (seeders)
- Currently incentives provided by central (closed) trackers
- See ***BarterCast*** for a fully distributed solution implemented in Tribler

Incentives for "indirect *reciprocity*":

- BT, like TFT, needs direct interactions between pairs: "you scratch my back and I'll scratch yours"
- But for some applications we need indirect reciprocity: "you scratch his back and I'll scratch yours"
- See ***GiveToGet*** for a distributed solution for Tribler video streaming

References

Nash, John (1950) Equilibrium points in n-person games. Proceedings of the National Academy of Sciences 36(1):48-49.

John von Neumann and Oskar Morgenstern: Theory of Games and Economic Behavior, Princeton University Press (1944)

Robert Axelrod (1984) The Evolution of Cooperation, Basic Books

Nowak, M.A. and Sigmund, K. (1998) Evolution of indirect reciprocity by image scoring, Nature 393, 573.

Garrett Hardin (1968) The Tragedy of the Commons Science 162, 1243-1248.

Trivers, R.L. (1971). The evolution of reciprocal altruism. Quarterly Review of Biology. 46: 35-57

Maynard Smith, J. (1982) Evolution and the Theory of Games. Cambridge University Press

References

J.A. Pouwelse et al (2008). Tribler: A social-based peer-to-peer system. Concurrency and Computation: Practice and Experience 20:127-138.

J.J.D. Mol et al (2008). Give-to-Get: Free-riding-resilient Video-on-Demand in P2P Systems. In Multimedia Computing and Networking 2008, San Jose, USA. SPIE Vol. 6818.

Michel Meulpolder et al (2008) Fully Distributed Sharing-Ratio Enforcement in BitTorrent, PDS technical report PDS-2008-002