# Self-Organising, Open and Cooperative P2P Societies – From Tags to Networks [1]

David Hales

Department of Computer Science, University of Bologna, Italy.
dave@davidhales.com

**Abstract.** For Peer-2-Peer (P2P) networks to realize their full potential their nodes need to coordinate and cooperate, to improve the performance of the network as a whole. But this requires the suppression of self behavior in the form of (free-riding). Existing P2P systems often assume that nodes will behave altruistically, but this has been shown to be far from the case (creating inefficient systems). We outline encouraging *initial* results from a P2P simulation that translates and applies the properties of Tag models (Hales 2000, 2001) to tackle these issues. We find that a simple node rewiring policy, based on the tag dynamics, quickly eliminates free riding between selfish nodes without centralized control. The process appears highly scalable and robust.

## 1 Introduction

Open Peer-to-Peer networks (in the form of applications on-top of the internet) have become very popular for file sharing applications (e.g. Kazaa[2], Gnutella[3] etc). However, as has been shown (Adar and Huberman 2000) in such file sharing scenarios we find that a majority of users do not actually share their own files (they act selfishly). However, these networks are still popular because it only requires a minority to share high quality files for all to benefit - a small amount of altruism appears to be enough to support file-sharing applications. But what about P2P applications where high levels of altruism and cooperation *are* required (e.g. load balancing or cooperative routing)? How can selfish nodes to be discouraged? One solution is to have a closed system in which we can ensure that each node runs a particular peer application that is hard-coded to be cooperative. But

---

[1]This work partially supported by the EU within the 6th Framework Programme under contract 001907 (DELIS).

[2]The Gnutella home page: http://www.gnutella.com/

[3]The Kazaa home page: http://www.kazaa.com

this option precludes the benefits of open systems. In open systems the protocols are open so any node that understands the protocol can participate. This allows for truly decentralized control and freedom for innovation (new nodes with new kinds of behavior may enter the network). A desirable goal would be to have a network that could self-organize and adapt to a variety of tasks such that each node would benefit from the shared resources (bandwidth, processing, storage etc) that other nodes could offer.

In order to archive such a desirable goal, don't we just need some very cleaver nodes? Or to put it another way, can we archive this goal "simply" by programming the peer nodes appropriately. Unfortunately there are some fundamental contradictions that need to be confronted when attempting to formulate how desirable collective action can be produced in open systems. We have to deal with the fact that each peer cannot make arbitrary *a priori* assumptions about the behavior of other peers. The assumptions (we can't avoid all assumptions) need to be as general as possible without being useless. Historically these issues have been studied and theoretically formulated within the social sciences (particularly Sociology, Political Science and Economics) with application to human social systems. Of course it would be foolish to believe that a set of generally agreed assumptions (concerning human social behavior) that a majority of social scientists would subscribe to could *ever* exist. There are several reasons for this including the complexity of human systems, the changing nature of human social organizations and behavior, the essentially political status that ideas applied to human society tend to acquire (particularly when those ideas are used to justify social action) and the fragmented nature of social science methodology. However, in the context of a kind of "worst case" set of assumptions, for engineering nodes in a P2P, we argue for that nodes should be seen as:

- In the network for what they can get out of it – selfish not altruistic
- Modify their behaviors to maximize individual benefit
- Have no (or limited) knowledge about other peers and the network in general

These assumptions imply a further one. That peers have some mechanism of determining how much they are benefiting from the system. This obviously would depend on the task domain e.g. for file sharing it would be some measure of how quickly requested files were found and downloaded or for group computing it might be a measure of processing resource donated by others[4].

Given these assumptions one fundamental problem is how to ensure that common resources (the commons) are utilized unselfishly for the benefit of all (Hardin 1968). In the context of a P2P network we can view each peer as offering a set of commons

[4] It should be noted that in many real word task domains it is by no means clear what these measures might be. Certainly one can imagine situations in which no such simple measures could be determined. This would be particularly difficult for very delayed rewards.

resources. That is, peers through their actions, may offer resources to others or may not. Conversely peers make use of the resources offered by others. The fundamental problem is that given peers with the above assumptions under what conditions would peers converge towards sharing (benefiting all) as opposed to selfishly taking resources but offering none.

There are many possible ways to deal with these problems including the utilization of trusted 3rd parties, the generation and sharing of reputation information and behavioural strategies based on sanctions in future interactions (Axelrod 1984). However, in general such mechanisms demand high overheads in form of storage, processing and communication of information concerning on-going interactions and / or do not work in highly dynamic contexts where interactions will be predominantly with strangers. In its most condensed and abstracted form these kinds of scenario can be captured in the two player, single round Prisoners' Dilemma game where players represent peers (see below).

Tags (see below) have recently been applied in these latter kinds of scenarios in the form of social simulations (with associated sociological interpretations - see Sigmund and Nowak, 2001). Firstly we will review relevant findings from the previous Tag simulations. Then we describe a simple simulation model of a P2P network and give some encouraging initial results. Finally we discuss the limitations of the model and the future direction we might take in order to address those limitations.

Along the way we attempt to present the method by which techniques have been imported from one kind of simulation scenario to another. The focus of the previous models was not on solving engineering problems; neither did those models deal with networks so the translation process was not straightforward.

## 2. What are Tags?

Tags are markings or social cues that are attached to individuals (agents) and are observable by others (Holland 1993). They evolve like any other trait in a given evolutionary model. The key point is that the tags have no direct behavioral implication for the agents that carry them. Through indirect effects (such as biasing of interaction), however, they can evolve from initially random values into complex ever changing patterns that serve to structure interactions between individuals.

In the computational models presented here tags are modeled using some number (either a binary bit string, a real number or an integer). When agents interact they preferentially interact with agents possessing the same (or similar) tag value. One way to visualize this is to consider a population of agents partitioned between different colors. Each agent carries a single color. In a system with only 3 different possible tag values we could think of this as each agent carrying a flag of either red, green or blue. Agents then preferentially interact with agents carrying the same color (forming 'interaction groups"). When agents evolve (using some form of evolutionary algorithm) they may mutate their tag (color). This equates to moving between interaction groups.

In the models presented here, tags take on many possible unique values (by say using a real number, there are many possible unique tags rather than just 3 colors) however, the basic process is the same – agents with the same tags preferentially interact and tags evolve like any other genotypic trait.

Another way to think of tags is that some portion of the genotype of an agent is visible directly in the phenotype but the other agents. In section 5 we give an outline algorithm of how tags are applied in a simple evolutionary systems, firstly however, we over the Prisoners Dilemma game and some previous tag work.

## 3. The Prisoner's Dilemma

The Prisoner's Dilemma (PD) game captures a scenario in which there is a contradiction between collective and self-interest. Two players interact by selecting one of two choices: Either to "cooperate" (C) or "defect" (D). For the four possible outcomes of the game players receive specified payoffs. Both players receive a reward payoff (R) and a punishment payoff (P) for mutual cooperation and mutual defection respectively. However, when individuals select different moves, differential payoffs of temptation (T) and sucker (S) are awarded to the defector and the cooperator respectively. Assuming that neither player can know in advance which move the other will make and wishes the maximize her own payoff, the dilemma is evident in the ranking of payoffs: $T > R > P > S$ and the constraint that $2R > T + S$. Although both players would prefer T, only one can attain it. No player wants S. No matter what the other player does, by selecting a D move a player ensures she gets either a better or equal payoff to her partner. In this sense a D move can't be bettered since playing D ensures that the defector cannot be suckered. This is the so-called "Nash" equilibrium for the single round game. It is also an evolutionary stable strategy for a population of randomly paired individuals playing the game where reproduction fitness is based on payoff. So the dilemma is that if both individuals selected a cooperative move they would both be better off but both evolutionary pressure and game theoretical "rationality" selected defection.

For a detailed treatment of the PD, its relationship to social and evolutionary science and a serious, original and thought provoking analysis of the evolution of non-suboptimal behavior from selfish interactions see Heylighen F. (1992)[5].

---

[5] Also see information about he PD online at the wonderful "Principia Cybernetica Project website: http://pespmc1.vub.ac.be/PRISDIL.html

## 4. Previous Tag Models

There have been a number of tag models implemented previously. All demonstrate higher-than-expected levels of cooperation and altruism from seeming selfish individuals. All implement evolutionary systems with assumptions along the lines of the replicator dynamics (i.e. reproduction into the next generation proportional to utility in the current generation, no "genetic-style" cross-over operations but low probability mutations on tags and strategies during reproduction).

Riolo (1997) gave results of expansive and detailed studies applying tags in a scenario where agents played dyadic (pair wise) Iterated Prisoner's Dilemma games (IPD). Tags (represented as a single real number) allowed agents to bias their partner selection to those with similar tags (probabilistically). He found that even small biases stimulated high levels of cooperation when there were enough iterations of the game with each pairing.

In Riolo et al (2001) a tag model was applied to a resource-sharing scenario in which altruistic giving was shown to emerge. Agents were randomly paired (some number of times) and decided if to give resources or not. The decision to give was based on tag similarity mediated by a "tolerance gene" as well as the "tag gene" (both represented as real numbers). The utility to the receiving agent of any given resource was greater than to that of the giving agent. It was shown that if each agent was paired enough times in each generation and the cost / benefit ratio was low enough then high levels of cooperation were found.

In Hales and Edmonds (2003) tags were applied to a simulated robot coordination scenario producing high levels of cooperative help giving.

In Hales (2000) a tag model was applied to a single round PD. Again interaction was dyadic. Tags were represented as binary strings. Pairing was strongly biased by tag identity (rather than probabilistic similarity). In this model very high levels of cooperation were produced between strangers in the one shot game. A refinement of this model in Hales (forthcoming) showed how the same result could be produce with tags represented as real numbers so long as the probability of mutation being applied to the tag is higher than that applied to the strategy (by about one order of magnitude).

For the purposes of this paper we will now outline in a little more detail these latter models applied to the PD.

## 5. Tags and the PD

In Hales (2000) a model is presented of agents playing the PD in pairs in a population with no topological structure (other than tag based biasing of interaction). The mode is composed of very simple agents. Each agent is represented by a small string of bits. On-going interaction involves pairs of randomly selected agents playing a single round of PD. Agent bits are initialized at random. One bit is designated as the PD strategy bit: agents possessing a "1" bit play C but those possessing a "0" bit play D. The other bits represent the agents tag. These bits that have no direct effect on the PD strategy selected by the agent but they are observable by all other agents. Below is an outline of the simulation algorithm used:

*LOOP some number of generations*
*LOOP for each agent (a) in the population*
*Select a game partner agent (b) with the same tag (if possible)*
*Agent (a) and (b) invoke their strategies and get appropriate payoff*
*END LOOP*
*Reproduce agents in proportion to their average payoff*
*Apply, with low probability, mutation to tag and strategy of each reproduced agent*
*END LOOP*

Agents are selected to play a single-round of PD not randomly but based on having the same tag string. If an agent can find an individual with the same tag string as its own in the system it will play PD against that agent. If it cannot then it plays against some randomly chosen partner. Agents are reproduced probabilistically in proportion to average payoff they received (using roulette wheel selection).

Extensive experimentation varying a number of parameters showed that if the number of tag bits is high enough[6] (in this case we found 32 tag bits for a population of 100 agents to be sufficient with a mutation rate of 0.001 and PD payoffs of T=1.9, R=1, P=S=0.0001[7]) then high levels of cooperation quickly predominated in the population[8].

More interesting still, if all the agents are initially set to select action D (as opposed to randomly set) then the time required to achieve a system where C actions predominate is found to monotonically decrease as population size increases. This is an inverse scaling phenomena: the more agents, the better. Additionally the fact that the system can recover from a state of total D actions to almost total C actions (under conditions of constant mutation) demonstrates high robustness. The tag-based model

---

[6] In a more recent model Hales (forthcoming) we demonstrate that the requirement for many tag bits was because this effectively increased the mutation rate applied to the tag as a whole (since mutation was applied to each bit with the same probability as mutation was applied to the single strategy bit).

[7] P and S were set to the same small value for simplicity. If a small value is added to P (enforcing T > R > P > S) results are not significantly changed.

[8] If tags are removed from the model and pairing for game playing is completely random then the population quickly goes to complete defection (the Nash equilibrium for the single-round PD).

produces an efficient, scalable and robust solution – based on very simple individual learning methods (modeled as reproduction and mutation).

### 5.1 How Tags Work

We have described this model because it seems to offer up a method for achieving three important properties in a simple (PD) task domain: efficiency, scalability and robustness. But how do tags produce this seemingly magical result? The key to understanding the tag process is to realize that agents with the same tag strings can be seen as forming a sort of "interaction group". This means that the population can be considered as a collection of groups. If a group happens to be entirely composed of agents selecting action C (a cooperative group) then the agents within the group will outperform agents in a group composed entirely of agents selecting action D (a selfish group). This means that more agents will copy the behavior of cooperative groups than selfish groups. By copying the behavior and the tags of those who perform well, agents are essentially joining groups that are cooperative. However, if an agent happens to select action D within a cooperative group then it will individually outperform any C acting agent in that group and, initially at least, any other C acting agent in the population – here the T payoff is 1.9 where as the best a C acting agent can do is R = 1.

However, by others copying such an agent (i.e. the agent reproducing copies of itself) the group becomes very quickly dominated by D acting agents and therefore the relative advantage of the lone D acting agent is lost – the group snuffs itself out due to the interaction being kept within the group. So by selecting the D action an agent destroys its group very quickly (remember groups are agents all sharing the same tag string). Figure 1 visualizes this group process in a typical single run. Each line on the vertical axis represents a unique tag string. Groups composed of all C action agents are shown in light gray (Coop), mixed groups of C and D agents are dark gray and groups composed of all D are black.

The tag mechanism, then, precipitates a kind of "group selection" process i n which those groups which are more cooperative tend to predominate but still die out as they are invaded by mutant D acting agents. In a real sense the groups compete for resources despite the fact that evolution only occurs at the individual level and the agents don't even know they are in such a group. In this system, the agents don't die, just the particular groupings (based on sharing the same tag string) change. By constantly changing tag strings (by reproduction of those with higher utility) the agents produce a dynamic process that leads to high levels of C actions. In other words, the population as a whole contains a lot of cooperation occurring within a constantly changing system of groups, even though each agent is acting without any knowledge of the group structure and there is no central coordination of the groups. Typically cooperative interactions in the model reach over 90% of all interactions (over 100,000 cycles).

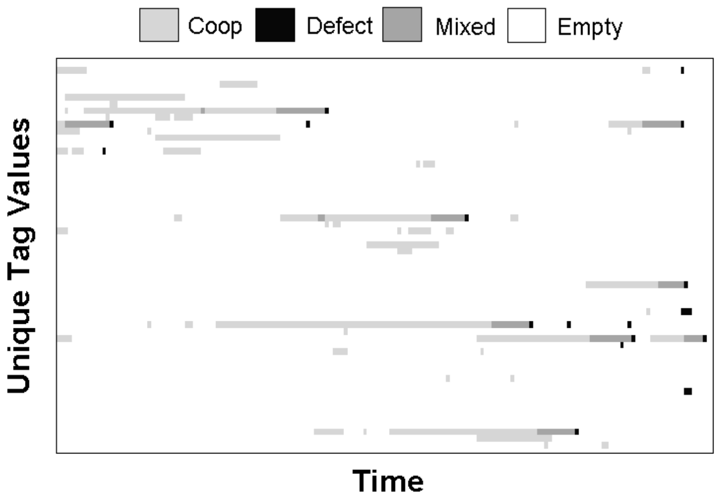


**Figure 1**. Visualization of 200 cycles (generations) from a single simulation run showing cooperative groups coming into and going out of existence. See the text for a full explanation

## 6. From Tags to Networks

The underlying mechanism driving cooperation within the tag simulation is the formation and dissolution of sharply delineated groups of agents (identified by sharing the same tag). Each agent could locate group members from the entire population. Each member of the group had an equiprobable chance of interacting with any agent in the population sharing the same tag. In this sense each agent could determine which agents were in their group.

If we assume a sparse P2P network in which each node (peer) knows of some small number of other nodes (neighbors) and those neighborhoods are highly interconnected (clustered) such that most neighbors share a large proportion of other neighbors then we have something similar to our tag-like groupings. Instead of a tag (a marker) we have an explicit list of neighbors. In a highly clustered network the same list will be shared by most of the neighborhood. In this sense you can visualize the table of known peers stored in each agent (its neighbors) as the tag.  It is shared by the group and is the key by which the group can directly interact with each other. To this extent it defines a group boundary. A nice feature of this also is that it is a kind of watertight method of isolating nodes into neighborhoods (for direct interaction) since a node cannot go directly interact with another node that it does not know of.

In our initial model we did not restrict the size of the neighborhood (i.e. networks could be non-sparse) but we wired the initial network topology as "small world" (i.e. highly clustered regular lattice but with random rewiring with low probability – see *watts ref). Also we set the tag mutation probability (changing the neighborhood to a single randomly chosen neighbor) to an order of magnitude higher than the strategy mutation probability (flipping the strategy bit). We found later that we did not need to wire a small world and that any initial wiring self-organized to high clustering over time. Sparse random wiring was finally chosen for simplicity.

We investigate only direct interactions between neighbors in this model. In a sense this is all that can ever happen in P2P systems. Indirect interactions between nodes that do not share neighborhoods have to mediate by direct interactions between intermediate nodes. Essentially, one can view all interaction as with neighbors (even if those neighbors are actually proxies for other more remote nodes). If cooperation can be established between the majority of neighborhoods in a network then it follows that any pair of nodes in the network that are connected will have a good chance of being able to find a *path of cooperation* through the network.

In order to capture this kind of neighborhood interaction in the simplest possible way we have each node in the network play a single round of PD (see above) with a randomly chosen neighbor. No information is stored or communicated about past interactions and the topology is not fixed (see below).

### 6.1 Neighbor Lists as Tags, Mutation as Movement

In the tag model change was produced over time by mutation and differential reproduction based on average payoff. How can these be translated into the network?

In our network model we do not view nodes as "reproducing" in a biological sense or cultural sense. However, it is consistent with or initial assumptions (see above) that nodes may relocate to a new neighborhood in which a node is performing better than itself. That is, we assume that periodically nodes make a comparison of their performance against another node randomly chosen from the network[9]. Suppose node (i) compares itself to (j). If (j) has a higher average payoff than (i) then (i) disregards its neighbor list and copies the strategy and neighbor list of (j) also adding (j) into the list. This process of copying can be visualized as movement of the node into the new neighborhood that appears more desirable.

---

[9] Currently we do not model the process of finding this "out-group" node. We assume that the network could provide the service – but this might be a problem (see conclusion).

Mutation in the tag model was applied after reproduction. Each bit of the tag and the strategy was mutated (flipped) with low probability. Since we are using the same one bit strategy we can apply mutation to the strategy in the same way. We therefore flip the strategy bit of a node with low probability immediately after reproduction (the movement to a new neighborhood as described above). Since we treat the list of neighbors in each node as the tag a mutation operation implies changing the list is some way. But we can't simply randomly change the list; we need to change the list in such a way as to produce an effect with closely follows what happens when mutation is applied in the tag model. In that model, tag mutation tended to give agents unique tags – i.e. tags not shared by other agents at that time. However, in the model agents could interact with a randomly chosen agent with non-matching tags if none existed with identical tags. In this way tag mutation lead to the founding of new tag groups. In the network model we don't want to isolate the node completely from the network otherwise it will not be able to interact at all. However, we don't want to move into an existing neighborhood (as with reproduction) but rather to do something that may initiate the founding of a new neighborhood. So we pragmatically express tag mutation as the replacement of the existing neighbor list with a single neighbor drawn at random from the network.

We now have our analogues of reproduction and mutation for the network model. Reproduction involves the nodes copying the neighbor lists and strategies of others obtaining higher average scores. Mutation involves flipping the strategy with low probability and replacing the neighbor list with a single randomly chosen node with a low probability. In the next section we outline out new network model – NetWorld[10].

## 7. The NetWord model

The NetWorld model is composed of a set N of nodes (or peers). Each node stores a list of other nodes it knows about (we term this the neighbor list). In addition to the neighbor list each node stores a single strategy bit indicating if it is to cooperate or defect in a single round game of the PD. Neither the strategy bit nor the list is normally visible to other nodes. Initially nodes are allocated a small number of neighbors randomly from the population. Periodically each node selects a neighbor at random from its list and plays a game of PD with it. Each node plays the strategy indicated by its strategy bit. After a game the relevant payoffs are distributed to each agent. Periodically pairs of randomly chosen nodes (i, j) compare average payoffs. If one node has a lower payoff then the strategy and neighbor list from the other node is copied (effectively moving the lower

[10] There are many other ways tags could be translated into networks. For example, agents could move around the network between nodes carrying tags or agents sharing a node could be seen as sharing a tag. We hope to explore some of these variations in the future.

scoring node to a new neighborhood). Mutation is applied to both the strategy and the neighbor table with probability m. Mutation of the strategy involves flipping the bit. Mutation of the neighbor list involves clearing the list and replacing it with a single randomly chosen node from the population. Below is an outline of the NetWorld simulation algorithm:

*LOOP some number of generations*
*LOOP for each node (i) in the population*
*Select a game partner node (j) randomly from neighbor list*
*Agent (i) and (j) invoke their strategies and get appropriate payoff*
*END LOOP*
*Select N/2 random pairs of agents (i, j) reproduce neighbor list and strategy of higher scoring agent*
*Apply mutation to tag and strategy of each reproduced agent with probability m*
*END LOOP*

The neighbor lists are limited in size to a small number of entries. The entries are symmetrical between neighbors (i.e. if node (i) has an entry for node (j) in its list then node (j) will have node (i) in its list). If a link is made to a node that has a full neighbor list then it discards a randomly chosen neighbor link in order to make space for the new link. Also if a node is found to have no neighbors when attempting to play a game of PD (this can happen if neighbors have moved away) then a randomly chosen node is made a neighbor.

### 7.1 Initial Results

Figure 2 gives some initial results. In these experiments the mutation rate m=0.001 and the PD payoffs were as per the previously described tag model (see above). In all the results given here we start he population from complete defection and wired the initial network topology by given each note a fixed small number of links (4) to randomly chosen nodes . We tried increasing the mutation rate applied to the tags (i.e. the neighbor list) by an order of magnitude and this reduced the time to cooperation an increased the scalability. Over all sensible parameter values so far tried we have found extremely encouraging results. Of particular surprise was the speed of convergence to high cooperation. Even when N=105 and all nodes were initially started with D (defect) strategies it took only approximately 140 cycles on average to achieve 99% of nodes utilizing the C (cooperate) strategy.

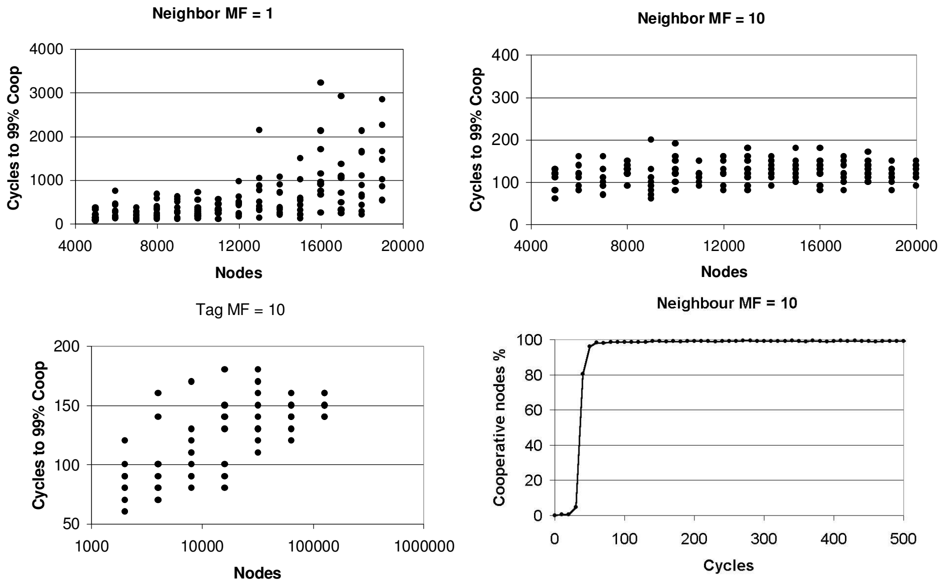


**Figure 2**. Charts show initial results from the model. The top charts show the time taken in cycles for the network to reach a state where 99% of nodes are cooperators for different sizes of network. Each dot is an individual run. In the right chart mutation on the neighborhood (tag) is 10 times that on the strategy. The bottom left chart shows extended results on a log scale. The bottom right chart shows a typical single run time series (with a 10,000 node network).

We hoped to find the reverse scaling cost and differential mutation characteristics identified in previous non-network based tag models (Hales 2000, forthcoming). The reverse scaling does not appear to be present. The application of differential (higher) mutation on tags (neighbor tables), appears to bring the upper-bound (of time to cooperation) down to log(nodes) improving scalability. As stated previously, we did not find that the initial form of the network made much difference to the results. Even starting the network fully connected or completely unconnected (all with no links) did not change our results significantly. This seems to suggest a high level of robustness – something we are interesting in achieving. Again much more experimentation and analysis is needed.

## 8. Discussion and Related Work

Only after our network translation of tags as the dynamical rewiring of the network (as nodes seek to improve their neighborhoods) did we realize the wealth of material that becomes relevant. Specifically our model now bears a very close comparison with that given by Zimmermann et al 2001. Zimmermann et al start with a network that is interpreted as representing a social network (it's a socia l simulation). Agents play PD with all their neighbors at each time step. Defecting agents then selectively replace existing links to other defectors with randomly chosen nodes (this is done probabilistically). They find steady states (of high cooperation) in which long chains of cooperators are formed in which founder 'leader" nodes are highly important to stability. Prior to steady states there are oscillations in the levels of cooperation. They use synchronous updating throughout and do not include noise (in the form of spontaneous change of strategy) in most of their analysis. However they do study the effect of a single noise event on significant nodes (they call leaders) and show that in their model even a single mutation event (changing the strategy of a single node) can completely wipe out cooperation in the entire network for many cycles. This is one of the major findings of the model, that the steady states become highly sensitive to changes in single key nodes. They also find that their network when in the steady state tends to be highly disconnected leading to a number of component sub-networks forming. In their model the network was populated with 60% cooperative agents. Since they have no spontaneous adaptation or noise on strategies their model would never escape from a global minima of all nodes defecting. So although the models are very similar theirs has several properties that we would want to avoid in our model[11] such as not being able to recover from total defection and the high sensitivity of the network to the behavior of a very small number of 'leaders". Additionally in their model agents needed a little more local information (i.e. the strategies of the other agents so they could preferentially break links).

The fitness related preferential rewiring in our model obviously has linkages with the preferential linking ideas expressed by Barabási (2002). In subsequent experimentation with our model we hope to characterize the kinds of networks that are being formed over time. It will be of interest to compare these to the kinds of naturally occurring networks and rewiring methods that have been studied by Barabási. We hope that we may be able to utilise these theoretical and empirical contributions to increase understanding and efficient of our model with respect to tougher task domain.

It should be mentioned that Watts (1999) looked at the results of playing the repeated PD game (rather than the single round game) on various fixed network topologies. Reproduction of strategies was within the local neighborhood. Various repeated

---

[11] However, we found it intriguing to consider if we would have reached a similar model to NetWorld if we had *started* with the Zimmermann et al model and attempted to make it more robust.

strategies were tested. On the whole no the simulation results presented showed that it was difficult to get cooperation to dominate the network even with repeated strategies like tit-for-tat as popularized by Axelrod (1984). He found that some kinds of fixed small world networks could help sustain cooperation.

Interestingly Cohen et al (1999) examined the results of extensive experimentation where *both* tags and networks (fixed random) were examined for their contribution to promoting cooperation in a PD scenario. Again only the *repeated* game was examined (not the single round game). However, they did not *combine* tags and networks but rather compared independent simulations.

A recent paper of Sun & Garcia-Molina (2004) applies "incentives" within a simulated P2P file-sharing scenario in order to encourage selfish nodes to share resource. Their model relies on repeated interaction with nodes updating weights between on links to neighbors. Although they have not yet tested their system in a evolving environment, they don't require utility comparison between nodes since nodes simply update their weights based on service gained and then share out service supplied proportionate to weight (a kind of tit-for-tat Axelrod (1984)). This means that a selfish node quickly gets less and less service from it's neighbors. In future work we hope to apply the query scenario given by Sun & Garcia-Molina to our more dynamic scenario.

## 9. Conclusion

At this early stage our conclusion contains more questions than answers. However, the basic result of these initial experiments is that high-levels of cooperation can be produced and sustained in very large P2P by following this simple re-wiring and mutation scheme inspired by results from previous tag models. It appears we have been successful in importing the tag like dynamics into the network.

As stated previously these are preliminary results from a preliminary model and there are a number if outstanding issues before we can refine the model to incorporate more realistic P2P-like conditions. For example, we don't model the maintenance of up-to-date neighbour tables in the face of unstable links and nodes. Neither do we model the underlying process of finding random nodes in the network. This shortcut needs to be modeled using the P2P itself to supply new such nodes for the purposes of reproduction and mutation. What would be important here would be to find an efficient scalable way (probably therefore non-uniformly random) to supply nodes that allowed cooperation to form. We hope to test our results on a simulated version of something like NEWSCAST (Jelasity et al 2004) - a highly robust and scalable P2P infrastructure.

We have yet to properly analyze the dynamics in the model. What kinds of networks topologies are being formed? We currently don't know how average path lengths, clustering and other topological features of the network evolve over time. It may even be the case the network regularly breaks into a number of disconnected components[12]. This would be serious problem if such breaks persist and are numerous since this would limit the possible size of the P2P network. All we currently know is that when cooperation is low the average degree of each node (size of the neighbor list) is near maximum but is lower when cooperation is high. This does not tell us too much.

The PD task domain although useful is a rather impoverished task domain. As an initial proof of concept it shows that at least some kinds of social dilemma can be solved. But the behaviors (PD strategies) and coordination required is trivial (although the dilemma itself is not trivial). We would therefore like to extend the simulation model to include more realistic kinds of task such as those requiring the coordination of a number of peers performing specialized functions.

A more important general issue raised by this kind of work[13] (in the context of applying models originating in the assumptions of evolutionary theory) is the assumption that all nodes behave as bounded optimizers. In our model we do not allow for nodes that simply "whitewash" (i.e. never adapt but just defect) or nodes that don't move, or worse nodes that move very fast but never adapt their strategy. This assumption does not hold in many situations and we need to explore alternative mechanisms to make model robust to these possibilities.

## Acknowledgements

Thanks go to Mark Jelasity for pointing out some of the recent models that bear close comparison to this one. Also, along with Ozalp Babaoglu, and Alberto Montresor, for writing clear and readable papers about P2P systems that have helped me in beginning this line of work. Thanks to Bruce Edmonds and Scott Moss at the Centre for Policy Modelling (CPM) in Manchester where much of the initial tag work was graciously supported and encouraged. Thanks also to the reviewers of the first draft of this paper *who really did* make valuable suggestions and significant pointers for future development of the model and were very generous given the very initial stages of the work presented.

---

[12] Very recent work (since the first review of this paper) does indeed show that the network regularly breaks into disconnected components – which raises issues of if this mechanism would support long range routing tasks. However, the network is in constant flux (in a similar way to the groups in figure 1) with cliques forming and dissolving so this may be possible over some temporal window.

[13] And pointed out by a perceptive reviewer of the initial draft of this paper!

## References

Albert-Lázló Barabási (2002) Linked: The New Science of Networks, Cambridge, MA: Perseus Publishing

Axelrod (1984) The Evolution of Cooperation, Basic Books, New York.

Davis, L.(1991) Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York.

Di Marzo Serugendo, G. "Engineering Emergent Behaviour: A Vision", Invited Talk. Multi-Agent-Based Simulation III. 4th International Workshop, MABS 2003 Melbourne, Australia, July 2003, D. Hales, B. Edmonds, E. Norling, J. Rouchier (Eds), LNAI 2927, Springer-Verlag, 2003.

Duncan Watts (1999) Small Worlds: The Dynamics of Networks between Order and Randomness. Princeton University Press. Princeton, New Jersey.

Edmonds, B. and Hales, D. (2003) Replication, Replication and Replication - Some Hard Lessons from Model Alignment. Journal of Artificial Societies and Social Simulation 6(4).

Eytan Adar and Bernardo A. Huberman (2000) Free Riding on Gnutella. First Monday Volume 5, No. 10, (http://www.firstmonday.dk/issues/issue5_10/adar/index.html).

Hales, D. (2000), Cooperation without Space or Memory: Tags, Groups and the Prisoner' s Dilemma. In Moss, S., Davidsson, P. (Eds.) Multi-Agent-Based Simulation. Lecture Notes in Artificial Intelligence, 1979:157-166. Berlin: Springer-Verlag.

Hales, D. (2001) Tag Based Cooperation in Artificial Societies. PhD Thesis (Dept. Of Computer Science, University of Essex, U.K. 2001).

Hales, D. (2002) Evolving Specialisation, Altruism and Group-Level Optimisation Using Tags. In Sichman, J. S., Bousquet, F. Davidsson, P. (Eds.) Multi-Agent-Based Simulation II. Lecture Notes in Artificial Intelligence 2581:26-35 Springer Verlag. Berlin.

Hales, D. (forthcoming), Change Your Tags Fast! - A necessary condition for cooperation? Submitted to the MAMABS workshop at AAMAS 2004.

Hales, D. and Edmonds, B. (2003) Evolving Social Rationality for MAS using "Tags", In Rosenschein, J. S., et al. (eds.) Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems, Melbourne, July 2003 (AAMAS03), ACM Press, 497-503

Hales, D. and Edmonds, B. (2004) Can Tags Build Working Systems? - From MABS to ESOA, In Di Marzo Serugendo, G.; Karageorgos, A.; Rana, O.F.; Zambonelli (eds.) Engineering Self-Organising Systems - Nature-Inspired Approaches to Software Engineering. Lecture Notes in Artificial Intelligence 2977, Springer, Berlin.

Hamilton, W. D. (1964) The genetical evolution of social behaviours, I and II. J. Theor. Biol. 7, 1-52.

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

Holland, J. (1993) The Effect of Lables (Tags) on Social Interactions. Santa Fe Institute Working Paper 93-10-064. Santa Fe, NM.

Heylighen F. (1992) : "Evolution, Selfishness and Cooperation", Journal of Ideas, Vol 2, # 4, pp 70-76.

Jelasity, M., Montresor,A., and Babaoglu, O. (2004) A modular paradigm for building self-organizing peer-to-peer applications. , In Di Marzo Serugendo, G.; Karageorgos, A.; Rana, O.F.; Zambonelli (eds.) Engineering Self-Organising Systems - Nature-Inspired Approaches to Software Engineering. Lecture Notes in Artificial Intelligence 2977, Springer, Berlin.

M. Cohen, R. Riolo, and R. Axelrod. (1999) The emergence of social organization in the prisoner's dilemma: how context-preservation and other factors promote cooperation. Santa Fe Institute Working Paper 99-01-002.

M. G. Zimmermann, Victor M. Egufluz and Maxi San Miguel (2001) Cooperation, adaptation and the emergence of leadership in `Economics with Heterogeneous Interacting Agents' , pp. 73-86, A. Kirman and J.B. Zimmermann (Eds.), Springer, Berlin.

Nowak, M. & May, R. (1992) Evolutionary Games and Spatial Chaos. Nature, 359, 532-554.

Nowak, M. & Sigmund, K..(1998) Evolution of indirect reciprocity by image scoring. Nature, 393, 573-557.

Riolo, R. (1997) The Effects of Tag-Mediated Selection of Partners in Evolving Populations Playing the Iterated Prisoner' s Dilemma. Santa Fe Institute Working Paper 97-02-016. Santa Fe, NM.

Riolo, R. L., Cohen, M. D. & Axelrod, R. (2001) Evolution of cooperation without reciprocity. Nature 414, 441-443

Roberts, G. & Sherratt, T. N. (2002) Nature 418, 449-500

Sigmund, K. and Nowak, A, M. (2001) Tides of Tolerance. Nature 414, 403-405.

Trivers, R. (1971) The evolution of reciprocal altruism. Q. Rev. Biol. 46, 35-57.

Qixiang Sun & Garcia-Molina (2004) SLIC: A Selfish Link-based Incentive Mechanism for Unstructured Peer-to-Peer Networks. In Proceedings of the 24th IEEE international Conference on Distributed Systems (March 2004). IEEE computer Society.