A Multi-agent Approach to Resource Sharing Optimization in User Networks

J.C. Burguillo-Rial, E. Costa-Montenegro, and F.J. González-Castaño

University of Vigo
ETSET. Campus Univ. de Vigo
36310-Vigo, Spain
{jrial, kike, javier}@det.uvigo.es

Abstract. In this paper, we evaluate the feasibility of multiagent control of resources to be shared in user networks. A user network is totally controlled by the users, both at application and transport level. One of the possible applications in these networks is peer-to-peer (P2P) file exchange sharing the "external" access to the Internet (set of links between the user network and the Internet). If a node cannot serve its demand with its own external link, it requests help from another node via the high-bandwidth internal user network. We model user nodes as agents to simulate and to evaluate a new agent-based distributed control scheme. The simulation results in this paper confirm that it is possible to improve resource sharing in user networks using agents that take decisions autonomously, from local information, and check that file exchange services offered to neighbour nodes do not surpass appropriate credit limits.

1 Introduction

User networks are totally controlled by the users, both at application and transport level. This paradigm has become possible with the advent of broadband wireless networking technologies such as IEEE 802.11. For applications such as peer-to-peer file exchange [6], it may be useful to consider the "external" access to the Internet (set of links between the user network and the Internet) as a shared resource that can be optimized by node cooperation (i.e., if a node cannot serve its demand with its own external link, it requests help from another node via the high-bandwidth internal user network).

In this paper, we analyze the conditions that enable cooperation in user networks. This is not trivial in realistic scenarios. We could impose conditions leading to resource sharing via node cooperation that would never hold considering real user behaviour in peer-to-peer (P2P) services. We model resource sharing for P2P file exchanges in user networks as a game where node routers can cooperate or defect. Defection models "free-rider" users [14]: a typical problem in P2P networks.

The main goals of this paper are (1) to show that agent-based cooperative nodes may become a majority in user networks, improving resource sharing, and (2) those agent-based nodes can learn easily how to avoid free-riders by using adaptive credits.

The paper is organized as follows. Section 2 introduces user networks, peer-to-peer systems and some basic concepts of Game Theory. Section 3 presents the multiagent scenario and finally section 4 draws the conclusions.

2 User-Managed Networks, P2P and Game Theory

User-managed networks have become possible with the advent of wireless technologies such as IEEE 802.11 [4]. They represent one of the last stages in network control evolution [5]. This kind of infrastructures is currently being used to provide broadband access in Spanish rural areas, as an expansion of shared asymmetric DVB-S or DVB-T gateways. A typical basic node in a wireless user-managed network is composed by a router, an IEEE 802.11 access point (AP) and/or some IEEE 802.11 cards to set links with other basic nodes. Basic nodes may also be linked to a multi-user LAN (covering a building, for example).

A subset of the basic nodes will have cable or DSL access, providing "external" connection to the Internet. For the purposes of this paper, we will assume that all basic nodes are "externally connected". Additionally, we assume that user network capacity is larger than external access capacity (this holds for reasonable internal and external networking technologies, for example IEEE 802.11 and DSL respectively), so that the internal network always has spare capacity. In a user network, basic nodes can easily share contents, due to the large internal bandwidth. The bottleneck is the set of "external" connections to the Internet. By optimizing their usage, overall performance (and, as a consequence, user satisfaction) can be greatly improved.

By network stability we refer to the condition such that external demands (e.g., downloads or uploads at the edges of the user network for peer-to-peer file exchanges) can be satisfied with external capacity, on average. This certainly holds if:

1. The external demand of each basic node can be satisfied with its own external capacity, on average.
2. All basic nodes cooperate via the user network and their combined external demand can be satisfied with their combined external capacity, on average.

While cooperation is not strictly necessary to guarantee network stability, cooperation minimizes demand service time (nodes with temporarily idle external connections can help neighbours with demand peaks). However, there is no central authority, and probably selfish nodes will act to maximize their performance.

In general the P2P model improves the capability for resource sharing in any type of network. Applications based on such model include file-sharing systems (such as eMule [11], Kazaa [8] or BitTorrent [9]), discussion boards [12], overlay routing [16], etc. In these types of systems cooperation can be a useful strategy but it consumes node resources and performance. Since rational self-interested peers always try to improve their performance, they can refuse to supply service to others when they do not have clear incentives. Then, the "Tragedy of Commons" [13] may appear leading peers to generalized defection, i.e., free-riding [14].

Game theory [7] provides useful mathematical tools to understand the possible strategies that self-interested agents may follow when choosing a course of action. The context of cooperative games and cooperation evolution has been extensively studied in biological, social and ecological contexts [1], seeking general theoretical frameworks like the Prisoner's Dilemma (PD). In his seminal work, Axelrod has shown that cooperation can emerge in a society of individuals with selfish motivations [2]. For a review of related work in the last twenty years see [3]. Game Theory and the Generalized Prisoner's Dilemma have been applied to solve incentive

problems in P2P systems. Examples can be found in [15] and BitTorrent [9] itself that considers an alternative of the Tit-for-Tat strategy [2].

3 Multiagent Scenario

In this section we present a multiagent approach to support node decision. Now, we consider nodes ruled by agents that take their own decisions concerning the strategy they follow and the maximum credit limitation (since now *CreditLimit*) they allow to the neighbouring nodes for using their external connection. Therefore, *CreditLimit* defines the maximum amount of help (i.e., data) an agent would provide to a peer without receiving help in return.

The approach we follow in this paper is a composite spatial game where actions are effectively simultaneous but every agent may interact with several neighbours at a time. Every agent receives a data throughput payoff every turn. We consider throughput payoff as the amount of data send plus data received per unit of time. The better the strategy in its context, the better the payoff, i.e., the higher the throughput. Note that the payoff of a given agent depends on the choices made by the rest. After a pre-defined set of turns (24 hours = one day), each agent i chooses a strategy $s_i \in S$, where S is the set of all strategies. The agent will keep the strategy chosen to interoperate with its neighbours for the next day.

Fig. 1. Node N_i and two possible neighbourhoods: first with 4 nodes including $\{N_{i1}, N_{i2}, N_{i3}, N_{i4}\}$ and second with 12 nodes, from N_{i1} to N_{i12}

3.1 Spatial Distribution

For the spatial distribution of the nodes we consider a two-dimensional square lattice consisting of N nodes. Each node will follow one of the basic strategies (defection or cooperation). Figure 1 shows a cell node i with a strategy s_i and two possible neighbourhoods. In the model discussed in the next section, every cell (i.e. node) in the square lattice will interact with its neighbours to handle external traffic. Thus, there are both local and spatial interactions between neighbour cells. If we let every node in the system to interact with the remaining $N-1$ nodes, we have a panmictic population. But, in real user networks each node interacts with only a few (3 to 4)

neighbours. Thus, we consider that each cell i only interacts with the 4 cells in its immediate neighbourhood ($Ni1$ to $Ni4$ in figure 1).

Interaction is driven by demand service times of externally inbound or outbound data, i.e., when the external service queue of a node is larger than a particular threshold, the node contacts its neighbours requesting help to handle the files involved. In order to introduce a time scale, the time unit to generate new traffic demands and for neighbours' interaction is an hour. The total number of interactions per hour can be $4 \times N$ or less.

We decided that, during a 24-hour timeframe (a day), the strategy s_i of a node does not change. We also decided to consider a day as the time unit for strategy changes, since traffic patterns are similar along different days.

3.2 Strategy Types

We model three different strategies as follows:

- *Defection*: a defector never helps but ask their neighbours for help when it needs it. A defector models again a free-rider.
- *Cooperation*: a cooperator always helps its neighbours without taking care about any limitation. Cooperation models altruist users in P2P networks.
- *Agent-based*: a node following this strategy will adapt individually and dynamically its *CreditLimit* value to optimize the interaction with its neighbours.

We implement help transactions using the Contract Net Protocol: neighbours answer with a set of offers and/or refusals. The requesting agent selects the offering node that provided the best average throughput in the past.

3.3 Network Traffic

Concerning node demand distribution, we now define three node types A , B and C . These node types generate distributed demand service times (externally inbound or outbound data) during disjoint busy and quiet periods. The parameters are:

- A nodes: The busy period runs from 0 to 8 hours with uniformly distributed demand service times with an average of 8 units. The quiet period has an average of 1 unit during the remaining 16 hours.
- B nodes: The busy period runs from 8 to 16 hours with an average of 8 units. The quiet period has an average of 1 unit during the remaining 16 hours.
- C nodes: The busy period runs from 16 to 24 hours with an average of 8 units. The quiet period has an average of 1 unit during the remaining 16 hours.

With these three node types we model three typical roles of Internet connection nodes: (A) for the late night users nodes (p.e., students' nodes), (B) for daily work users (p.e., commercial nodes); and (C) for leisure-time users (workers' home nodes). We also choose an average of 8 units during the busy period, which it is a little bit over the service time per hour (5 units) but not too much. The global service demanded per node in a day has an average of: $(8 \times 8 + 1 \times 16 = 80)$, which is less than $(24 \times 5 = 120)$. So we fulfil the two conditions described in section 2.

3.4 Simulation Scenario

We take a similar approach to the one presented in [10]. We consider that every agent stores a vector with the number of times NT_i that agent i has used every possible strategy. We define a strategy efficiency estimator to be:

$$EE_i(s, d+1) := \alpha \cdot f(i, d) + (1 - \alpha) \cdot EE_i(s, d) \quad (1)$$

Where $f(i, d) = \min_{h \in d}(th(h, i))$ represents, the minimum throughput value (th) obtained by agent i during any hour of that day. We consider the worst throughput as users try to improve their worst conditions. The α parameter is obtained from the formulae:

$$\alpha = w + (1-w) / NT_i(s) \quad (2)$$

Being $NT_i(s)$ the number of times that agent i uses the strategy s and w is a real-valued constant. We set ($w=0.3$) considering the results described in [10]. The term $(1-w)/NT_i(s)$ is a correcting factor, which has a major effect only when $NT_i(s)$ is low, when $NT_i(s)$ grows this term becomes negligible with respect to w .

To select the new strategy for the next day we need a probability distribution. Initially, we force every agent to test every possible strategy at least once. Then we do:

$$EN_i(s) = EE_i(s, d+1)^n \quad (3)$$

Where n is a positive real-valued parameter. To turn this into a probability we do for every strategy $s \in S$:

$$Prob_i(s) = EN_i(s) / ET_i \quad (4)$$

Being $ET_i = \sum_s EN_i(s)$ the normalization factor. Then clearly $EN_i(s)$ bias the selection towards strategies that have performed well in the past. The strength of the bias depends on n ; the larger the value of n , the stronger the bias. With high values of n (e.g., $n > 20$) the agent will always choose the strategy with the best record. But as explained in [15], this option does not allow the agent to explore other strategies when there are changes in the context. Therefore we set ($n = 10$).

3.5 Learning *CreditLimit*

Using agent-based strategy we want that every node learn what is the better *CreditLimit* it should apply considering its traffic conditions and the context where it is located, i.e., the surrounding neighbours and the interaction with them.

The agent has not too much time to explore the space of values of *CreditLimit*, since a delay in the convergence to the right value could cause throughput losses. In this framework, we consider that evolutive algorithms perform enough good and somehow simpler than other more sophisticated optimization techniques [17]. An evolutive algorithm considers a population that evolves on three phases: couple selection, crossover and sporadically mutation. This simple algorithm is defined as follows:

0. Every agent takes one random sample in every interval: $[1, 10]$, $[10, 20]$, $[20, 30]$, $[30, 40]$, $[40, 50]$.
1. The agent chooses the best values (CL_1 , CL_2) obtained and they become the selected couple. The other CL values are forgotten.

2. The newborn CL_3 is a linear combination (crossover) of its parents: $CL_3 = CL_1 + (1 - x) CL_2$ where $x = \text{rand}(0, 1)$.
3. Mutation: IF $(\text{rand}(0, 1) < 12 / \text{Hours})$ THEN $CL_3 = CL_3 + \text{rand}(-5, 5)$.
4. If CL_3 is better than CL_1 and CL_2 then the worst parent is replaced.
5. Return to step 2.

Note: In step 3, the first check is at 24 hours so the probability is lower than 1. We also limit the *CreditLimit* value range to $[1, 50]$ after considering bigger intervals that were not used by the agents but introduce search delays.

3.6 Simulation Results

In this section we present the results obtained in the simulations performed with the Java UserNet simulator developed by the authors (access can be freely provided on demand). We considered a cell lattice of $(25 \times 25 = 625)$ cells in the user network. We also considered equally probable the initial selection of the 3 types of nodes and the 3 strategies. Every node neighbourhood has 4 nodes as described in figure 1.

Figure 2 shows the frequency evolution of the strategies along 50 days. The agent-based strategy becomes the more popular followed by the defector one. Cooperators still survive linked or surrounded by agent-based nodes as can be seen in the left side of figure 3, where darker cells are defectors, lighter ones cooperators and middle-dark agent-based nodes. In fact, agent-based nodes give somehow a skeleton to group cooperators around them with defectors surviving isolated and mainly exploiting cooperators. The value of *CreditLimit* learnt by the agents, had an average of 20.8 (in 10 runs) with a variance of 11.2 and most of the values fall in the range $[5, 30]$.

Right side of figure 3 displays the throughputs achieved by the three strategies. At the beginning defectors have better results because they exploit cooperators and waste

Fig. 2. Evolution of cooperators (C), defectors (D) and agent-based nodes (A) after 50 days

Fig. 3. Left: cell lattice state. Right: throughput comparison of cooperators, defectors and agent-based nodes. Time: 50 days; cooperators (C), defectors (D) and agent-based (A).

the *CreditLimit* provided by the agent-based nodes. Then, after 400 hours (around 17 days) cooperators and mainly agent-based nodes get better. At the end defectors perform half-better than the others.

Therefore, learning nodes chose the agent-based strategy as the most effective to avoid defectors (i.e., free-riding) and to improve their daily throughput.

4 Conclusions and Future Work

In this paper we present an abstraction of the problem of resource sharing in user networks. We have abstracted the main features of a real user network to obtain a simpler yet valid simulator. Then we show that under certain conditions (like learning and setting proper credit limits by every agent) cooperation becomes the most popular strategy, even in case of fully isolated and distributed node operation.

We point out that although the model finally looks simple, it has been far from trivial to be obtained. Thus, besides the model, the main contribution of this paper is to show that agent-based strategy nodes in user networks may become a majority, improving resource sharing, and avoiding free-riders by using adaptive credits.

Concerning future work, we think that these results can be extended to consider classical P2P file sharing over the Internet. In the short future we plan to design and implement a new protocol layer in user-networks to enable communication among all the user nodes. With such protocol, the agent coterie will not be restricted to its immediate neighbourhood providing global addressing within the user network (like the Internet). We also want to consider more realistic distributions for traffic and user network topologies. Finally, we consider interesting to extend the model to consider mobile user networks with dynamic connection establishment.

References

- [1] F. Schweitzer, J. Zimmermann, H. Muhlenbein: Coordination of decisions in a spatial agent model. *Physica A*, 303(1-2), 189-216, (2002)
- [2] R. Axelrod: *The evolution of Cooperation*. Basic Books, New York, (1984).
- [3] R. Hoffmann: Twenty years on: The evolution of cooperation revisited. *Journal of Artificial Societies and Social Simulation*, 3(2), (2000).
- [4] IEEE 802.11. [Online]. Available at the web site: <http://grouper.ieee.org/groups/802/11/>
- [5] J.P. Hubaux, T. Gross, J.Y.L. Boudec, M. Vetterli: Towards self-organized mobile ad-hoc networks: the terminodes project. *IEEE Commun. Mag.*, 1, 118-124, (2001)
- [6] Kazaa news. [Online]. 2004. Available at the web site: <http://www.kazaa.com/us/news/index.htm>
- [7] Ken Binmore: *Game Theory*. Mc Graw Hill, (1994).
- [8] Kazaa participation ratio. [Online]. 2005. Available at the web site: http://www.kazaa.com/us/help/glossary/participation_ratio.htm
- [9] The official BitTorrent page, <http://www.bittorrent.com>
- [10] A. Schaerf, Y. Shoham, M. Tennenholtz: Adaptive Load Balancing: A Study in Multi-Agent Learning. *Journal of Artificial Intelligence Research*, 2, 475-500, (1995)
- [11] Y. Kulbak, D. Bickson: The eMule Protocol Specification. [Online]. 2005. Available at the web site: http://leibniz.cs.huji.ac.il/tr/acc/2005/HUJI-CSE-LTR-2005-3_emule.pdf
- [12] Gu. B, and Jarvenpaa. S.: Are Contributions to P2P Technical Forums Private or Public Goods? - An Empirical Investigation, In *1st Workshop on Economics of Peer-to-Peer Systems*, (2003).
- [13] Hardin, G. The Tragedy of the Commons. *Science* 162, 1243-1248, (1968)
- [14] E. Adar and B. A. Huberman, Free riding on Gnutella. (2002).
- [15] Michal Feldman, Kevin Lai, Ion Stoica and John Chuang, Robust Incentive Techniques for Peer-to-Peer Networks. *ACM E-Commerce Conference (EC'04)*, (2004).
- [16] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and DS Wallach, Security for Structured P2P Overlay Networks. In *Proceedings of Multimedia Computing and Networking*, (2002).
- [17] U.M. García-Palomares, F.J. González-Castaño, J.C. Burguillo-Rial. A Combined Global & Local Search (CGLS) Approach to Global Optimization. *Journal of Global Optimization (JOGO)*. (article in Press, beginning of 2006).