Cooperation in Decentralised Networks

Michael Rogers and Saleem Bhatti

University College London

Abstract: This paper describes a new model of cooperation between autonomous participants in decentralised networks. Our approach is based on reciprocation between immediate neighbours, which does not require any centralised infrastructure for accounting or identity management. We describe the conditions that are necessary for reciprocation to occur, and show how reciprocation between immediate neighbours can support multi-hop interactions such as packet forwarding.

1. Introduction

The problem of encouraging cooperation in decentralised networks has recently attracted considerable attention, particularly in the areas of peer-to-peer and mobile ad hoc networks. In these systems, the infrastructure and resources are provided collectively by the users, which can lead to a conflict of interests: it is in the interest of each user to consume the resources contributed by other users without contributing any resources in return. We consider this problem in terms of the economic concept of **utility-maximising behaviour**.

2. Description of the Model

Our model consists of a network of nodes, each connected to a number of other nodes which we call its neighbours. Nodes can request services from their neighbours and can measure the level of service received. Each node attempts to maximise the benefit it receives from its neighbours while minimising the cost it incurs by providing services.

Every action a node can perform has an estimated cost and one or more possible outcomes. Each outcome has an estimated benefit and an estimated probability of occurring. (The sum of the probabilities must be 1, i.e. each action has exactly one outcome.) Costs and benefits may be positive or negative, and they are subjective - it is never necessary to compare the estimates of one node to those of another node. We do not assume that costs are necessarily measured in the same units as benefits.

The **expected benefit** of an action is defined as the mean benefit of all possible outcomes, weighted by probability. An action's **expected utility** is defined as the expected benefit per unit of cost. When faced with a choice of actions, a **selfish node** is one that always chooses the action with the highest expected utility, according to its own estimates.

3. Reciprocation

A selfish node will not provide services unless is has an incentive to do so. The simplest form of incentive comes from **reciprocation**: if a node believes that providing a service to a neighbour will result in a higher level of service being provided in return, it can weigh the expected benefit of the reciprocation against the cost of providing the service.

To estimate the benefit of obtaining reciprocation from a neighbour, a node makes the assumption that all the benefit received from the neighbour so far is a result of reciprocation (in other words it assumes that the neighbour is selfish). This means that the benefit received from the neighbour can be attributed to the services provided to it. The total benefit received divided by the number of services provided tells the node the average benefit obtained by providing each service, which is the expected benefit of providing another service.

When evaluating a request in this way, a node does not know the benefit of the requested service from the neighbour's point of view - costs and benefits are subjective - but it can estimate the cost (to itself) of providing the service and the benefit (to itself) of the resulting reciprocation. Thus our model provides a quantifiable incentive for cooperation using only local information.

If it is possible to measure costs and benefits in the same units then the cost of obtaining reciprocation can be subtracted from the expected benefit, and a node may decide not to provide a service if the expected benefit is less than the cost. However, we do not assume that costs and benefits can necessarily be compared in this way. Instead, our approach depends on comparing the expected utilities of possible actions. By evaluating the expected utility of serving each request, a node can prioritise requests to maximise its own benefit by obtaining the most reciprocation per unit of cost.

4. Cooperation Over Longer Distances

Reciprocation can encourage selfish nodes to provide services to their neighbours, but in some networks the requester and provider of a service might not be neighbours. For example, a node that is asked to forward a packet in a mobile ad hoc network might not be a neighbour of either the original sender or the final recipient. To encourage cooperation in these situations, we must find a way of applying single-hop reciprocation to multi-hop interactions.

Our solution is to divide each multi-hop interaction into a series of single-hop interactions. Each node provides a service to its upstream neighbour without regard to whether that neighbour is the original requester, and requests a service from its downstream neighbour without regard to whether that neighbour is the final provider. The final provider creates a **proof of work** that can be verified by intermediate nodes as well as the requester. Each node returns the proof of work to its upstream neighbour, showing that the requested service has been performed. As before, reciprocation occurs only between immediate neighbours, but now there is an incentive to participate in multi-hop as well as single-hop interactions.

5. Examples

We offer two examples to illustrate how the model described above can be applied in practice.

5.1. A File Sharing Network

Imagine a file sharing network similar to BitTorrent [1]. Each node requests pieces of a file from its neighbours and counts how many bytes of the file have been provided by each neighbour. Benefit is measured in bytes downloaded and cost is measured in bytes uploaded. Since neighbours are assumed to be selfish, the amount downloaded from a neighbour can be attributed to the amount uploaded to it.

Nodes have a limited amount of upstream bandwidth, so a node may receive more requests from its neighbours than it is able to satisfy. A node maximises its benefit by serving requests in decreasing order of expected utility, regardless of the order of arrival. Expected utility is measured in bytes downloaded per byte uploaded, so the file is obtained at minimum cost by serving neighbours in strict priority according to their download/upload ratios.

5.2. A Mobile Ad Hoc Network

The second example concerns packet forwarding in a mobile ad hoc network. We assume that each node benefits from having its packets delivered, and wishes to save power by transmitting as few packets as possible. Benefit is measured in bytes delivered and cost is measured in bytes transmitted, including packets forwarded for other nodes.

Proof of work in this scenario is provided by a new technique which we call **delivery receipts**. The source and destination use a shared key to calculate a **unique secret** for each packet. The source hashes the unique secret using a one-way hash function (e.g. SHA1 [2]), and attaches the hash to the packet. Intermediate nodes store the hash before forwarding the packet. When the destination receives the packet, it releases the unique secret as a delivery receipt. Intermediate nodes can verify the receipt by hashing it and comparing the result to the stored hash. The one-way nature of the hash function ensures that only the intended recipient can create a valid receipt; intermediate nodes do not need to share any keys with the source or destination. The receipt is forwarded upstream to the source, proving to each node that its downstream neighbour performed the requested service.

A node might forward a packet that is later lost downstream, in which case it will have performed the service requested by its upstream neighbour but will be unable to prove it. To guard against this possibility, each node should take the reliability of the downstream path into account when calculating the expected benefit of forwarding a packet. (Recall that expected benefit is defined as the mean benefit of all possible outcomes, weighted by probability.)

6. Requirements for Reciprocation

The approach we have described is suitable for any network that meets the following requirements:

1. Authentication. A node must be able to verify that the neighbour requesting a service is the same neighbour that has provided services in the past. The necessary authentication can be achieved by exchanging ephemeral public keys when two nodes first make contact - no public key infrastructure is required[1].

2. Repeated interaction. The value of reciprocation comes from the expectation of future interactions with the same neighbour, so the expected benefit of reciprocation is lower if connections between neighbours tend to be short-lived. Uncertainty about the lifetime of a connection can be incorporated into the model by applying a discount factor to future benefits [3]. Very high churn in a peer-to-peer network or very high mobility in a wireless network could mean that connections between neighbours do not last long enough to make reciprocation worthwhile.

3. Proof of work. A node must be able to measure the benefit provided by each neighbour. Multi-hop interactions require a proof of work that can be verified by intermediate nodes as well as the original requester.

7. Related Work

Previous approaches to cooperation in decentralised networks can be divided into three categories: micropayments, reputations, and reciprocation. Micropayment systems require either tamper-resistant hardware [4] or a central bank [5] to prevent double spending. Reputation systems [6, 7, 8, 9, 10] require limits on the creation of new identities, otherwise nodes can discard tarnished identities or submit false positive reports about themselves. It is not generally possible to prevent a node from using multiple identities without central coordination [11]. In contrast, reciprocation [1, 12] requires only local information.

Urpi *et al.* [3] present a formal model of selfish behaviour in mobile ad hoc networks, and examine the effect of mobility on cooperation. In their model, a node benefits from the delivery or further forwarding of the packets it forwards, regardless of their origin. In our model, a node only benefits from the delivery of its own packets.

Felegyhazi *et al.* [13] use game theory to study the emergence of cooperation in mobile ad hoc networks. Each node treats the rest of the network as a single opponent in the repeated prisoner's dilemma [14], without distinguishing between neighbours. This approach has the advantage that no authentication is required, but it is vulnerable to exploitation by strategic nodes.

Simulations by Lai et al. [15] show that cooperation can emerge in a peer-to-peer network where each node keeps a private record of its interactions and adapts its behaviour towards unknown nodes based on past experience.

Feldman and Chuang [16] show that incentives conditional on end-to-end delivery can support multi-hop packet forwarding.

8. Conclusion and Future Work

We have outlined a new approach to cooperation in decentralised networks, which uses the concept of utility-maximising behaviour to model reciprocation between selfish nodes. Our model is quite general: we treat costs and benefits as subjective and do not assume that costs are necessarily measured in the same units as benefits. We briefly discussed the application of our model to file sharing and mobile ad hoc networks, and described delivery receipts, a new proof of work scheme for multi-hop packet forwarding.

Evolutionary simulations. The next step is to perform simulations of the file sharing and packet forwarding scenarios, comparing the benefit obtained by reciprocators with the benefit obtained by free riders, and measuring the overall benefit (social welfare) that results from various proportions of altruists, reciprocators and free riders. Evolutionary simulations will show whether reciprocation is an evolutionarily stable strategy [14].

Dynamics. Serving requests in order of expected utility could lead to complex dynamic behaviour: in general, serving a neighbour will decrease its priority, while receiving service from a neighbour will increase its priority. Simulations will allow us to explore the resulting dynamics and their possible effects on higher protocol layers.

[1] An attacker might be able to intercept the initial key exchange and insert itself between two nodes that would otherwise have been neighbours (a man-in-the-middle attack). However, a selfish node has nothing to gain by doing so: the victims will treat the attacker like they would treat any other new, presumably selfish neighbour. Any scheme that involves selectively forwarding requests between the victims can also be carried out by connecting to both victims as an ordinary neighbour.

Bootstrapping. We also need to consider the problem of establishing reciprocal relationships. When two nodes first make contact, one or both of them must take a small risk and provide a service without knowing how much reciprocation (if any) will result. The expected benefit of first-time interactions could be set artificially high in order to bootstrap reciprocal relationships at the risk of a one-time cost [14], or it could be based on the average benefit obtained from previous first-time interactions [15]. To discourage nodes from exploiting first-time cooperation by continually changing identities, puzzles [17] could be exchanged during handshaking.

References

[1] B. Cohen. *Incentives Build Robustness in BitTorrent.* Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, June 2003.

[2] D. Eastlake and P. Jones. *RFC 3174: US Secure Hash Algorithm 1 (SHA1).* September 2001.

[3] A. Urpi, M.A. Bonuccelli and S. Giordano. *Modelling Cooperation in Mobile Ad Hoc Networks: a Formal Description of Selfishness.* Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '03), Sophia-Antipolis, France, March 2003.

[4] L. Buttyan and J.P. Hubaux. *Nuglets: A Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks.* Swiss Federal Institute of Technology Technical Report, January 2001.

[5] L. Anderegg and S. Eidenbenz. *Ad Hoc VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents.* ACM Mobicom, 2003.

[6] S. Marti, T.J. Giuli, K. Lai, and M. Baker. *Mitigating Routing Misbehavior in Mobile Ad Hoc Networks.* 6th Annual ACM/IEEE Int. Conference on Mobile Computing and Networking, Boston, MA, USA, August 2000.

[7] Z. Despotovic and K. Aberer. *Maximum Likelihood Estimation of Peers' Performance in P2P Networks.* 2nd Workshop on Economics of Peer-to-Peer Systems, Cambridge, MA, USA, June 2004.

[8] S. Buchegger and J.Y. Le Boudec. *A Robust Reputation System for P2P and Mobile Ad Hoc Networks.* 2nd Workshop on Economics of Peer-to-Peer Systems, Cambridge, MA, USA, June 2004.

[9] A. Blanc, Y.K. Liu, and A. Vahdat. *Designing Incentives for Peer-to-Peer Routing.* 2nd Workshop on Economics of Peer-to-Peer Systems, Cambridge, MA, USA, June 2004.

[10] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. *Sustaining Cooperation in Multi-Hop Wireless Networks.* 2nd Symposium on Networked System Design and Implementation, Boston, MA, USA, May 2005.

[11] J.R. Douceur. *The Sybil Attack.* In Proc. 1st Int. Workshop on Peer-to-Peer Systems (IPTPS '02), Lecture Notes in Computer Science 2429, pages 251-260, Springer, 2002.

[12] Q. Sun and H. Garcia-Molina. *SLIC: A Selfish Link-Based Incentive Mechanism for Unstructured Peer-to-Peer Networks.* 24th Int. Conference on Distributed Computing Systems, 2004.

[13] M. Felegyhazi, J.P. Hubaux, and L. Buttyan. *Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks.* To appear in IEEE Transactions on Mobile Computing.

[14] R. Axelrod. *The Evolution of Cooperation.* New York, Basic Books, 1984.

[15] K. Lai, M. Feldman, I. Stoica, and J. Chuang. *Incentives for Cooperation in Peer-to-Peer Networks.* Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, June 2003.

[16] M. Feldman and J. Chuang. *Hidden-Action in Multi-Hop Routing.* 2nd Workshop on Economics of Peer-to-Peer Systems, Cambridge, MA, USA, June 2004.

[17] C. Dwork and M. Naor. *Pricing via Processing or Combatting Junk Mail.* In Proc. 12th Annual Int. Cryptology Conference, Lecture Notes in Computer Science 740, pages 139-147, Springer, 1992.