

Analysis of a feedback-based energy conserving content distribution mechanism for mobile networks

Mikael Lindberg*

* *Lund University, Sweden (e-mail: mikael.lindberg@control.lth.se).*

Abstract: A feedback-based scheme for energy conservative cooperative content distribution for mobile systems is presented together with analysis of its behavior in simulated operation. The mechanism described is designed to address problems of co-operation in transient groups, where group members are assumed to act selfishly to preserve their limited resources. Properties of the resulting barter-trade like economy are discussed as well as design rules when designing exchange systems of this type.

Keywords: Feedback, design, mobile networks, content distribution, energy

1. INTRODUCTION

Mobile networking is growing rapidly, taxing the infrastructure to its limits. Studies show that licensed spectrum is already congested and that something needs to be done to meet future demands of mobile network users. Of particular relevance is digital media distribution, as the traffic generated by video and music services already make up a significant part of bandwidth usage (Wolfson et al., 2007; Moustafa et al., 2012).

Peer-to-peer assisted mechanisms (Androutsellis-Theotokis et al., 2004; Kreitz and Niemela, 2010) are commonly used in wired content distribution schemes to off load network links and central repositories, but are rarely seen in the mobile case out of concern for the energy consumption involved. The transient nature of ad-hoc wireless networks as well as the risk of running out of energy before accumulated goodwill can be capitalized upon, makes cooperative schemes risky and therefore unattractive.

Because sharing data with other parties incurs an energy cost for both sender and receiver, it is vital to such a scheme that the party sharing the data can expect to have a return on the invested energy. The risk of being cheated, either by malicious intent or due to changes in the network population, effectively turns the proposition into the classic game theoretical example called The Prisoner's Dilemma. A system aimed at enabling cooperation in this setting must therefore explicitly address this risk, which is done in this paper by introducing a mechanism for enforcing agreements between clients.

Feedback based decision mechanisms are necessary in exchange systems where the client population fluctuates over time. Since this is one of the defining characteristics of mobile networks, the algorithms presented in this paper will rely only on information that is easily obtainable at runtime.

The scenario used as a motivating example is the distribution of files to a population of mobile clients, for

example a firmware upgrade or a new version of a popular application. The clients involved are assumed to have a primary connection to a remote service offering this file, reachable through an expensive long distance link (e.g. 3G or 4G), and the capability for local connections with co-located clients using cheaper communication forms (e.g. WLAN or Bluetooth).

2. RELATED WORKS

Peer-to-peer schemes have been considered for mobile applications before, though initially as a method for content distribution in ad-hoc networks or mesh networks (Kortuem et al., 2001; Ding and Bhargava, 2004). Resource conservation is not the primary focus in these works, but rather the availability of content.

Energy and spectrum concerns are introduced in (Shen et al., 2005; Yaacoub et al., 2012; Wolfson et al., 2007), but assumes that individuals are willing to cooperate without guaranteed return on energy expenditure. As such, these works sidestep the game theoretical issues that could lead to poor interest in participating in such schemes.

Fairness is explicitly mentioned in (Yaacoub et al., 2012), which also exploits the difference in cost between long and short range communication. The article employs game theoretical reasoning, but primarily as a method to solve the optimization problem rather than to discuss the issue of trust and willingness to participate. The solution presented in this paper is primarily an off-line approach, making it difficult to apply in cases with dynamic client populations.

Analysis of barter-trade economies is done across a variety of fields, including kidney exchanges and apartment contract trading (Marin and Schnitzer, 2002; Ashlagi et al., 2011). While similar in their graph theoretical reasoning, a significant difference is that clients in these systems are primarily interested in a single commodity or object, while in the case of peer-to-peer content distribution clients collect complete sets of specific objects that are in themselves worthless without the others in the set. In these systems,

there is often no alternative to cooperation, such as a central repository of goods.

3. AVOIDING THE PRISONER'S DILEMMA

Conserving energy is a primary concern for most mobile clients. The risk of expending energy without guaranteed benefit will be a strong deterrent from cooperating with other clients, as modeled by the Prisoner's Dilemma.

Consider two geographically co-located mobile clients, A and B, that both seek to fetch a specific data object from a remote service. Assuming the cost of short range communication, such as Bluetooth, is low compared to that of long range, such as 3G or LTE, the clients could potentially cooperate and share the cost of remote access.

Let w_l denote the long range download cost for one quantity of data and w_s the cost of transferring the same quantity across the short range link and furthermore that $w_l \gg w_s$. Transferring data requires energy expenditure by both sender and receiver, meaning that one local transfer incurs a cost of w_s for both parties.

If for instance A starts to download the data while unconditionally sharing it with B, A runs the risk of not benefitting from arrangement, as B could move out of short range communication range or simply decide not to share any data in return, resulting in the scenarios described by the table below.

Cost for (A,B)	B cooperative	B uncooperative
A cooperative	$(w_l + 2w_s, w_l + 2w_s)$	$(2w_l + w_s, w_l + w_s)$
A uncooperative	$(w_l + w_s, 2w_l + w_s)$	$(2w_l, 2w_l)$

If both cooperate, the costs will consist of one repository access to fetch half the data and then two short range accesses, one to share the data with the peer and one to fetch the complementary data from the same.

In game theoretical terms, this game has a Nash equilibrium where both A and B choose to be uncooperative (Axelrod, 2006). This outcome could be avoided if the off-diagonal choices were eliminated, thereby guaranteeing that cooperative agreements would always be honored. A solution would be to introduce a trusted 3rd party, who shares a stake in the outcome of the game. Logically, this would be the remote service provider, that would benefit in terms of reduced network congestion. A prototype mechanism is described in (Lindberg, 2013).

The question of whether or not to allow multicast transfers, i.e. letting clients not part of the agreement listen in, is problematic. This would improve the overall energy efficiency of the scheme, as shown by (Wolfson et al., 2007; Yaacoub et al., 2012), but there is the risk that clients will choose to only listen to multicasts, essentially re-introducing a Prisoner's Dilemma like situation. The proposed mechanism is therefore based on uni-cast communication only.

4. SYSTEM MODEL

Consider a population of mobile clients capable of multi-mode communication, that is, able to use several wireless

communication standards. Specifically, they support both an expensive long range communication mode and a less expensive short range alternative. It is assumed that all clients in this system are within short range communication distance of each other, for instance inside a commuter bus or a class room.

Let $C = \{c_i, i = 1..N_c\}$ denote the set of clients, and let $D = \{d_j, j = 1..N_d\}$ denote the set of data objects that constitute the file that all clients seek to download. In order to simplify notation, it is furthermore assumed that all objects are of equal size.

All parts of D can be accessed over long range link from a central repository. The nominal cost for a client to download the entire file is thus $N_d w_l$, which the client seeks to reduce by instead sharing downloaded parts with peers.

The set of clients C , the target data set D and the remote service constitute an Exchange System $E = (C, D)$, with the objective of allowing clients to minimize their data retrieval costs. The state of the system is the contents of the client side caches, i.e. the data objects a client has collected.

In this paper E is realized as a centralized mechanism with complete knowledge of the system state. The system dynamics evolve in discrete time steps of indeterminate length, but with the following logical sub-steps *discovery*, *arbitration* and *effectuation* that are repeated in a loop.

- **Discovery.** During discovery, clients join the exchange system and submit their current state. Let $\kappa(c)$ be a function that returns the data objects currently possessed by the c and $cardinal(\kappa(c))$ a function that returns the number of elements in $\kappa(c)$.
- **Arbitration.** Once the system state is established, the exchange system decides which trades that will occur.
- **Effectuation.** Finally all decisions are carried out. The completion of these actions marks the end of the time step, after which the next step immediately starts with a new discovery phase. Clients not part of a trade will perform a default action, that can be either fetching an object from the central repository or passing (i.e. doing nothing).

Clients can join or leave the exchange system at all times, either voluntarily (e.g. having completed the data set or user command) or involuntarily (e.g. by moving out of short range communication distance), making repeated discovery necessary. A client can choose to pass in the Effectuation phase rather than immediately download data from the remote service, as coming time steps might provide opportunities for trade. The tendency to do so is modeled by the parameter *skipcount*, which describes how many time steps a client is willing to wait for a trade opportunity.

5. THE BARTER TRADE ECONOMY

Keeping the economy healthy requires that the effect on future trade opportunities is always considered. Given that each client will only want to trade for each object once and that they will leave the exchange system once the complete

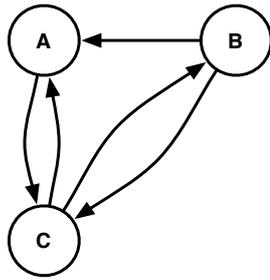


Fig. 1. A swap graph for the system $A[1, 2], B[1], C[3]$

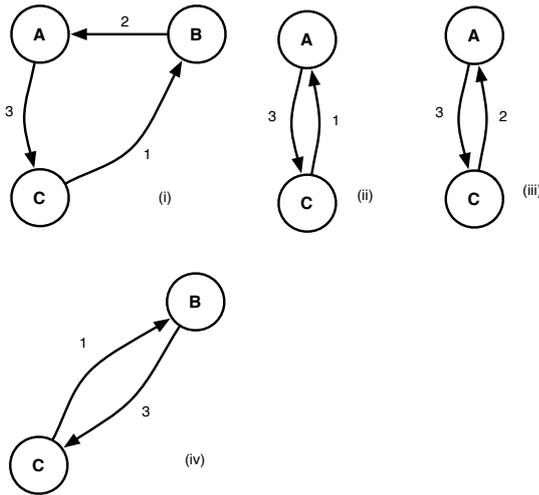


Fig. 2. The cycles given by the swap graph in Figure 1. Each of the cycles are mutually exclusive, meaning only one of them can take place. Arbitrating this conflict is a responsibility of the exchange system.

set is acquired, the system should promote trades that generate more trade opportunities for all clients.

5.1 The swap graph

To further discuss the properties of these systems, the concept of the swap graph will be used. This is a directed graph representation of which trades are possible, where each vertex represents a client and each edge represents a potential object transfer. If client A has an object desired by client B , then the swap graph will contain an edge from B to A , labeled with the object in question, to indicate the dependency. As there can be multiple dependencies between two clients, there can be multiple edges but with different labels.

As an example, consider a case with the client set $\{A, B, C\}$ and the data object set $\{1, 2, 3\}$. In the example, let client A possess objects 1 and 2, represented by the short hand notation $A[1, 2]$. Assume now that the total system state is $A[1, 2], B[1], C[3]$. The corresponding swap graph is seen in Figure 1. Possible fair exchanges are seen as cycles in the graph, with in total four in the example, as detailed in Figure 2. In this case they are mutually exclusive, which leads to the central question of arbitration, that is determining what exchanges should take place in order to optimize the objectives.

In its entirety, the problem is a multistep decision problem, where in each step determining the set of exchanges to

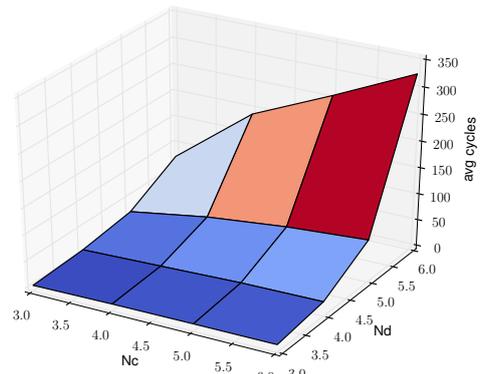


Fig. 3. The number of cycles grows very fast with the dimensions of the system. This plot shows the average number of cycles over 50 simulations when all clients are assigned randomly chosen states.

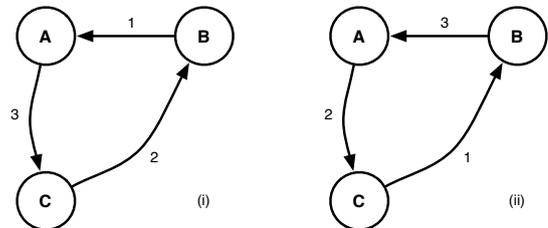


Fig. 4. Swap graphs for the special case discussed in Section 5.2. The system is initialized with the state $A[1], B[2], C[3]$, as shown in Graph (i), which allows for a 3-way exchange involving all clients, leading to the situation depicted in Graph (ii).

perform involves finding the best set of non-overlapping cycles in the graph, a version of the classic NP-hard maximum set packing problem (Karp, 1972). The computational complexity in each step grows as $2^{N_{cycles}}$ and since the number of cycles grows very fast with set sizes, as shown in Figure 3, finding the globally optimal solution is intractable for realistic scenarios involving hundreds of clients. However, a possible key to alternative strategies presents itself by studying a special case of $N_c = N_d$.

5.2 $N_c = N_d$

Consider a case with $N_c = N_d = 3$, with the system state $A[1], B[2], C[3]$ and the corresponding swap graph shown in Figure 4-i. After trivially selecting the exchanges, the state becomes $A[1,3], B[1,2], C[2,3]$, with the swap graph in Figure 4-ii. This gives another trivial decision that ends the scenario (as all clients are done) with an optimal cost for all clients, $N_d w_l + 2(N_d - 1)w_s$.

The two following observations can now be made:

- (1) The trivial optimal strategy above is always possible when all clients have the same number of objects and every object occurs the same number of times in the system.

- (2) The solution is not unique in general, there might be many other ways to achieve the same optimal global cost.

Let

$$n_c = \text{cardinal}(\kappa(c))$$

and

$$f_d = \sum_{c \in C} I_d(c)$$

where $I_d(c)$ is an indicator function defined as

$$I_d(c) = \begin{cases} 0 & \text{if } d \notin \kappa(c) \\ 1 & \text{if } d \in \kappa(c) \end{cases}$$

Furthermore, let \bar{n} denote the average number of objects possessed by clients in the system and \bar{f} denote the average object frequency, calculated as

$$\bar{n} = \frac{1}{N_c} \sum_{c \in C} n_c \text{ and } \bar{f} = \frac{1}{N_d} \sum_{d \in D} f_d$$

Using this notation, the condition from Observation 1 can be formalized into

$$n_i = n_j, \forall i, j \in C \text{ and } f_k = f_l, \forall k, l \in D \quad (1)$$

from here on referred to as Condition A.

Consider now the function

$$J = \sum_{c \in C} (n_c - \bar{n})^2 + \sum_{d \in D} (f_d - \bar{f})^2 \quad (2)$$

The quantity J can be seen to denote the distance to A, or if it is assumed that the optimal trajectory will be followed once A is fulfilled, the distance to the optimal trajectory.

The quantities $\sum_{c \in C} (n_c - \bar{n})^2$ and $\sum_{d \in D} (f_d - \bar{f})^2$, essentially the sample variance of the client cache sizes and object frequencies respectively, can be interpreted to model two aspects of how well the exchange system will work.

If the cache size variance is high, then some clients will finish way ahead of others, thereby removing many objects from the system. It therefore makes sense to prioritize clients with few objects when arbitrating exchanges.

When frequency variance is high, some objects are rare and few clients can offer them, while some are frequent, meaning few clients want them. Both cases lead to fewer possible exchanges involving these objects. It therefore makes sense to try to keep object frequencies uniform.

5.3 Analysis of J 's impact on trading

To further demonstrate the correlation between the quantity J and number of trading opportunities, Figure 5 shows the result of a Monte-Carlo type simulation where a system state was generated 5000 times by randomly placing $N_d/2$ objects among N_c clients, for $N_d = N_c = 8$. For each random state, the swap graph was constructed and the number of cycles, i.e. the number of possible trades, were plotted against the corresponding value of J .

The negative impact on possible trades for high values of J is clear but for smaller values of J , the correlation grows weaker. It can therefore be expected that an algorithm based on minimizing J will primarily work to prevent particularly bad trades, but that it will perform more or less on par with picking trades at random when the system is close to optimum.

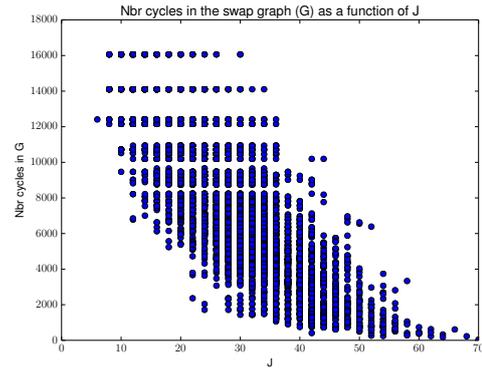


Fig. 5. The correlation between J and the number of cycles in the swap graph G . Note how high values of J drastically limits the number of cycles, while the correlation grows weaker as J approaches 0.

6. BASELINE ALGORITHM

Using Equation (2), it is possible to formulate a one-step decision algorithm based on minimizing J . Basing decisions only on the currently measurable state of the system, in this case the contents of the client side caches, is a feedback control approach. This has the advantage of being robust to disturbances, such as failed transfers or clients arriving to or departing from the exchange system. A pre-calculated multistep decision strategy would, on the contrary, have to be recalculated if for instance the state of the system suffers an unforeseen perturbation, such as a failed object transfer or clients leaving E .

Let X denote the system state, u denote a set of exchange agreements to carry out and $J(X|u)$ denote the cost function evaluated for the state after X has been subjected to u . Furthermore, let $\rho(X)$ be a function that maps the system state to a set of possible exchange agreements. The feedback arbitration policy can now be written as

$$u = \arg \min_{u \in \rho(X)} J(X|u) \quad (3)$$

Because of the combinatorial nature of the optimization problem used to calculate (3), designing the function $\rho()$ is non-trivial. The formulation is very close to the maximum set packing problem and as discussed in Section 5.1, the number of possible decisions grow unmanageably large even for modestly sized problems. However, it can still be useful to compare other solvers with the result given if $\rho()$ is assumed to generate all possible agreements, from now on referred to as the *baseline algorithm*.

7. HEURISTIC SOLVER

An heuristic alternative to (3) is to try to generate cycles that reduce J , although perhaps not in an optimal manner. A steepest decent style algorithm for cycle finding has been developed and is presented in detail in (Lindberg, 2013).

7.1 Simulation examples

The performance and behavior of the algorithm has been studied through simulations.

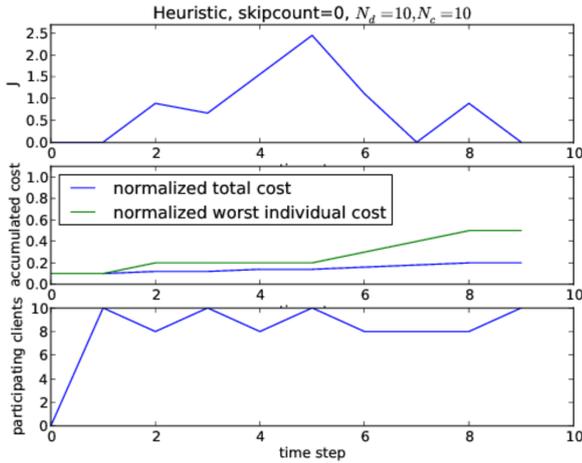


Fig. 6. System trajectories for the example in Section 7.1. The costs have been normalized so that a cost of 1 corresponds to the worst case cost, that is the case where all objects are fetched from the remote repository. For the individual cost the normalization factor is $1/(N_d w_l)$ and for the total cost the factor is $1/(N_c N_d w_l)$.

Figure 6 shows a scenario with 10 clients with skipcount 0 and 10 data objects, where all clients start out empty. The cost w_l is unitary for simplicity and w_s is assumed to be sufficiently small so that it can be approximated to 0. The figure shows how J , the total accumulation cost for the entire system and the worst case individual client cost evolve over time until all clients have completed their data sets.

As the initial state satisfies Condition A, the optimal trajectory is known and would give a worst case individual cost of one remote access, resulting in a normalized individual cost of $1/N_d = 0.1$, and a normalized total cost also of $1/N_d = 0.1$. The solver is not able to achieve this, but reduces the total cost by 90% and the worst individual cost with 60% compared with the non-cooperative case.

The scenario is repeated in Figure 7 with a skipcount of 2. The resulting cost is closer to optimal, but at the expense of taking more time steps to achieve.

Figure 8 shows the worst case individual cost for different combinations of N_d and N_c (using the heuristic solver and a skipcount of 10 in all cases). It can be seen that the level jumps approximately each time N_d crosses a multiple of N_c , a phenomena further discussed in (Lindberg, 2013).

From this it can be concluded that

$$\lceil N_d/N_c \rceil \quad (4)$$

is a reasonable predictor for the worst case individual number of remote accesses using this solver. Using (4) as a predictor for the number of repository accesses, the cost to download N_d objects can be written as

$$\lceil N_d/N_c \rceil w_l + 2(N_d - \lceil N_d/N_c \rceil) w_s \quad (5)$$

Assuming w_s is low enough to be negligible and normalizing with the nominal cost of $N_d w_l$, the predicted normalized cost under the mechanism proposed in this paper is $\lceil N_d/N_c \rceil / N_d$, meaning that given $N_c \geq N_d$, the

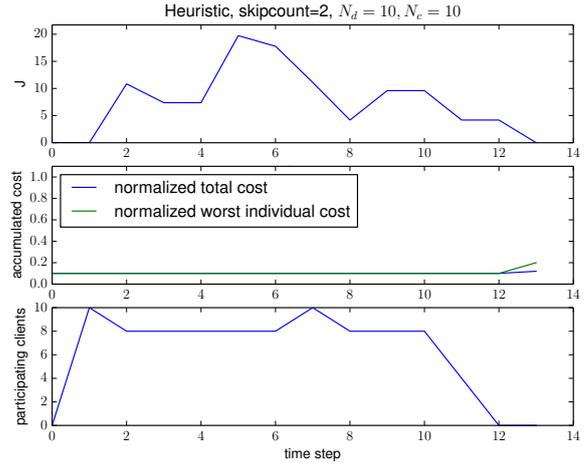


Fig. 7. System trajectories for the example in Section 7.1, with the skipcount parameter set to 2, resulting in lower individual and total costs at the expense of more time steps. Only a single client is forced to do two repository accesses, which is seen in the slight increase of worst case cost in the final time step.

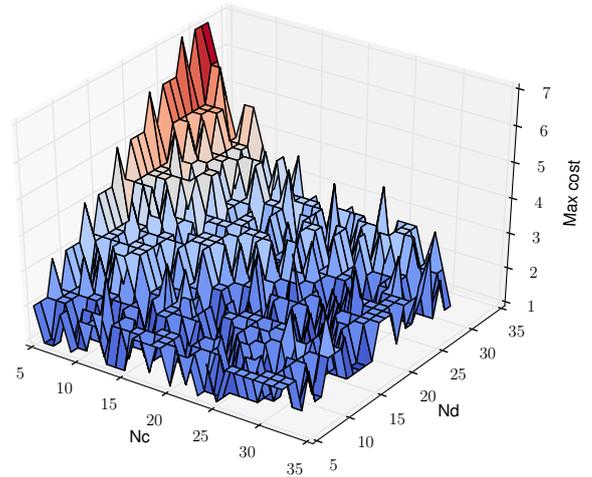


Fig. 8. The worst case individual costs (non-normalized) for scenarios of various sizes, all using skipcount of 10 which has experimentally been proven sufficient for all the cases in this simulation to reach their lowest costs. w_l is set to 1 (i.e. the cost is the same as the number of repository accesses).

cost scales with $1/N_d$. As such the predicted energy savings are on par with results presented in (Yaacoub et al., 2012).

The main advantage of the feedback method presented in this paper is that it does not require off-line optimization and is therefore able to handle continuous operation under random loads, as seen in Section 8.

8. CONTINUOUS OPERATION

The approach presented in this paper is primarily targeted at scenarios where the population will change over time, thereby making pre-calculated solutions unviable. A performance evaluation was therefore carried out using

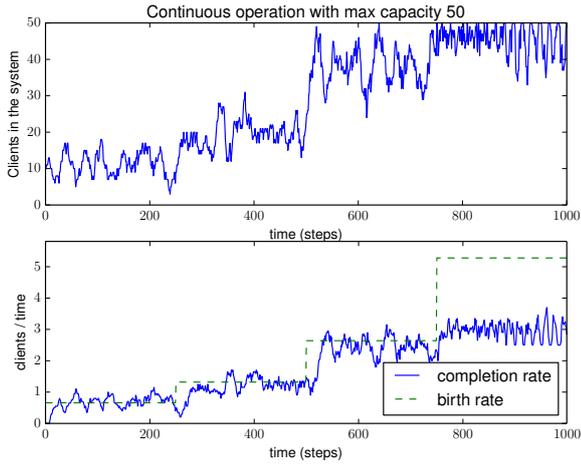


Fig. 9. Simulation of an exchange system with $N_d = 16$ in continuous operation with new clients arriving through a Poisson process with increasing birth intensity. The system becomes congested when the intensity goes above 3. The completion rate is calculated over a 20 time steps long sliding time window.

simulated scenarios for a case where $N_d = 16$ and with new clients arriving to the system through a Poisson process. The birth intensity was then made to increase over time to see how the system behaves under varying load. All clients in the simulation are modeled to have a skipcount of 10, meaning they are willing to accept some latency in order to save energy.

Once a client has completed its data set, it will leave the system. The exchange system has a hard limit for the maximum number of clients it will accept and clients arriving when the system is full will simply be denied access and removed from the simulation.

8.1 System capacity and congestion

Figure 9 shows how a system with a max capacity of 50 clients behaves under increasing arrival rate. The congestion point is very clear, the completion rate will not go above 3. The reason for this can be understood through the problem decomposition properties discussed in (Lindberg, 2013). As the average cost in this scenario is close to 1, the system performs nearly optimally. Therefore, the clients can be considered to group up in clusters of size N_d , where each cluster will complete its data set within N_d time steps. Each such cluster will therefore result in an average completion rate of 1 and with a maximum system capacity of 50, there can be $\lfloor 50/16 \rfloor = 3$ such clusters.

By increasing the maximum capacity of the system, more clusters can be formed and this will improve the throughput. Figure 10 shows how the completion rate will continue to match the birth intensity if the max capacity is increased to support at least 6 complete groups.

Figure 11 shows what happens around the congestion point in more detail. In order to improve visibility, the birth process is in this case deterministic, i.e. a specific number of new clients arrive each time step. As before, the system becomes congested at a birth rate of 3 clients per time

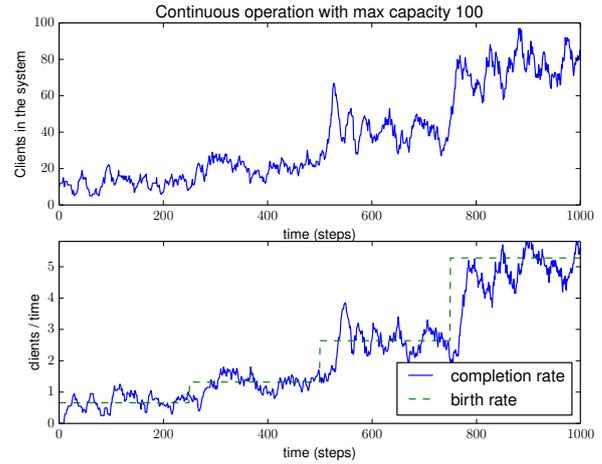


Fig. 10. Simulation of an exchange system with $N_d = 16$ in continuous operation with new clients arriving through a Poisson process with increasing birth intensity. The system remains uncongested as intensity goes well above 3. The completion rate is calculated over a 20 time steps long sliding time window.

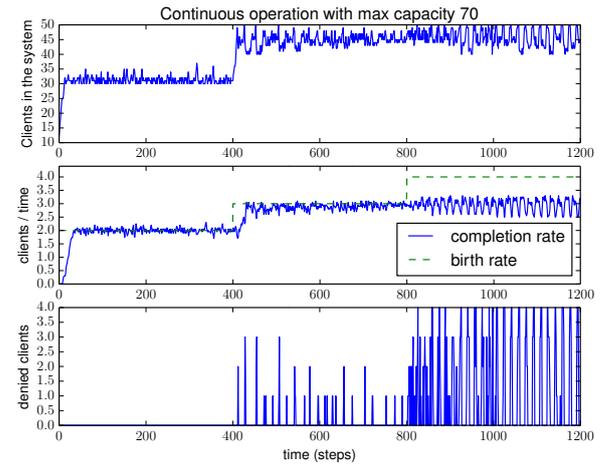


Fig. 11. Simulation of an exchange system with $N_d = 16$ in continuous operation with new clients arriving at a deterministic rate that increases over time. The system becomes congested as intensity reaches 3, resulting in some clients being denied. As birth rates go even higher, the number of clients being denied increases, which reduces the inflow of rate objects to the economy causing fluctuations in completion rate. time window.

step. The oscillations in completion rate once the system becomes congested can be explained by considering the effect of the default client actions. Because newly arrived clients will fetch the least common object from the remote service in order to have something to trade with, a steady inflow of clients will keep the variance in object frequencies low. When clients are denied, the inflow of rare objects is reduced which forces clients to wait more often for beneficial trades, thereby reducing system throughput.

8.2 System design

Given this knowledge about when the system becomes congested, it is possible to relate some of the system design parameters to each other

$$\begin{aligned}t_{step} &\geq \frac{S(D)}{rN_d} \\t_c &\leq t_{step}N_d \\b_{max} &\leq \lfloor \frac{N_{max}}{N_d} \rfloor \\c_{tot} &= w_l + 2(N_d - 1)w_s\end{aligned}$$

where t_{step} is the length of a time step in seconds, $S(D)$ the size of the complete file in bits, r the lowest bit-rate used to transfer data either locally or remotely, t_c the time to collect the complete file, b_{max} the maximum arrival rate to be serviced, N_{max} the maximum number of clients in the system and c_{tot} the complete cost of collecting the file for each participating client.

To illustrate how to utilize these design rules, consider a case where the objective is to dimension a system to be used in a commuter bus. Assume for this case that the iteration time is set to 1 minute and the bus has maximum capacity of 50 people of which 40 are assumed to be participating in the exchange system. Selecting how to divide a file, e.g. an operating system update, depends on minimum data rate, desired energy savings and how quickly the bus population changes. For maximum cost savings, the N_d should be as high as possible. However, this reduces system throughput. For maximum throughput, N_d should be as low as possible, but this reduces cost savings. If the passenger turnover in the bus is about 4 persons / minute, then $N_d \leq 10$.

9. CONCLUSIONS AND FUTURE RESEARCH

This paper has shown how a feedback based algorithm for co-operative content distribution can be used to reduce energy expenditure and licensed spectrum usage for a system of mobile clients. The concept of a barter trade market has been introduced together with a heuristic algorithm to optimize the health of the economy. The results of applying these principles have been demonstrated in simulated scenarios, together with a result showing how the throughput of the system depends on the design parameters.

Going forward, the algorithm should be modified to account for non-synchronous system dynamics. Furthermore, it is desirable to formulate a non-centralized decision algorithm to facilitate co-operation in mesh network situations. Finally, it is possible to extend the economy to include more diverse trades, for instance trading a data object for a sensor reading (e.g. a GPS measurement) using the energy needed to perform services as the base of the economy.

10. ACKNOWLEDGEMENTS

This work has been partially funded by the Lund Center for Control of Complex Engineering Systems (LCCC) and the Excellence Center at Linköping - Lund on Information Technology (ELLIIT).

REFERENCES

- Androutsellis-Theotokis, Stephanos, and Spinellis, D. (2004). A survey of peer-to-peer content distribution technologies. *ACM Computing Survey*, 36(4), 335–371. doi:10.1145/1041680.1041681. URL <http://doi.acm.org/10.1145/1041680.1041681>.
- Ashlagi, I., Gamarnik, D., Rees, M.A., and Roth, A.E. (2011). The need for (long) chains in kidney exchange. National Bureau of Economic Research. Market Design Working Group Meeting.
- Axelrod, R. (2006). *Evolution of cooperation*. Basic Books.
- Ding, G. and Bhargava, B. (2004). Peer-to-peer file-sharing over mobile ad hoc networks. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*.
- Karp, R. (1972). Reducibility among combinatorial problems. In R. Miller, J. Thatcher, and J. Bohlinger (eds.), *Complexity of Computer Computations*, The IBM Research Symposia Series, 85–103. Springer US. doi:10.1007/978-1-4684-2001-2_9. URL http://dx.doi.org/10.1007/978-1-4684-2001-2_9.
- Kortuem, G., Schneider, J., Preuitt, D., Thompson, T., Fickas, S., and Segall, Z. (2001). When peer-to-peer comes face-to-face: collaborative peer-to-peer computing in mobile ad-hoc networks. In *Proceedings of the First International Conference on Peer-to-Peer Computing*, 75–91. doi:10.1109/P2P.2001.990429.
- Kreitz, G. and Niemela, F. (2010). Spotify – large scale, low latency, p2p music-on-demand streaming. In *Proceedings of the 10th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 1–10. doi:10.1109/P2P.2010.5569963.
- Lindberg, M. (2013). Feedback-based cooperative content distribution for mobile networks. In *Proceedings of the 8th Performance Monitoring, Measurement and Evaluation of Heterogenous Wireless and Wired Networks Workshop*.
- Marin, D. and Schnitzer, M. (2002). *Contracts in Trade and Transition: The Resurgence of Barter*, volume 1. The MIT Press, 1 edition. URL <http://EconPapers.repec.org/RePEc:mtp:titles:0262133997>.
- Moustafa, H., Maréchal, N., and Zeadally, S. (2012). Mobile multimedia applications: Delivery technologies. *IT Professional*, 14(5), 12–21. doi:10.1109/MITP.2012.46.
- Shen, H., Joseph, M., Kumar, M., and Das, S. (2005). Precinct: A scheme for cooperative caching in mobile peer-to-peer systems. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, 57a–57a. doi:10.1109/IPDPS.2005.353.
- Wolfson, O., Xu, B., and Tanner, R.M. (2007). Mobile peer-to-peer data dissemination with resource constraints. In *Proceedings of the International Conference on Mobile Data Management (2007)*.
- Yaacoub, E., Al-Kanj, L., Dawy, Z., Sharafeddine, S., Filali, F., and Abu-Dayya, A. (2012). A utility minimization approach for energy-aware cooperative content distribution with fairness constraints. *Transactions on Emerging Telecommunications Technologies*, 23(4), 378–392. doi:10.1002/ett.1546. URL <http://dx.doi.org/10.1002/ett.1546>.