The Dangers of Poorly Connected Peers in Structured P2P Networks and a Solution Based on Incentives

Björn-Oliver Hartmann, Klemens Böhm, Andranik Khachatryan, Stephan Schosser
Universität Karlsruhe (TH), 76131 Karlsruhe, Germany
{hartmann|boehm|khachat|schosser}@ipd.uni-karlsruhe.de

Abstract

This paper analyzes structured P2P systems where peers choose both their interaction mode, i.e., how they process incoming queries, and additional contacts in the network autonomously. Since additional contacts incur additional costs, a new kind of free riding behavior, namely having only few contacts, comes into the fray. We refer to it as ***deliberately poor connectedness (dpc)****. In this paper, we show that dpc is dominant in many situations. This leads to networks with a low degree of connectivity and a higher overall forwarding load than necessary. We then propose an incentive mechanism against dpc and demonstrate its effectiveness using a formal analysis and experiments.*

1 Introduction

Structured Peer-to-Peer systems (P2P systems) allow to administer large numbers of data objects. With most structured P2P systems, there is a fixed topology: Each peer is connected to certain other peers, its *neighbors*. This paper investigates structured P2P systems where peers can establish connections to further peers based on utility considerations, i.e., not predefined by any topology. The neighbors and these further peers are the *contacts of a peer*. A peer can choose peers as additional contacts that it deems cooperative. A network of peers with a high degree of cooperation results, and lost messages are unlikely. In this paper, a connection between peers is bilateral: Both peers can use it and have to pay for it.

In unstructured P2P systems, peers tend to behave selfishly [18]: They try to benefit from the system without contributing. We expect this effect in structured P2P systems as well. Since the connection costs a peer must take depend on its number of contacts, it is rational to establish only few additional contacts if any. A peer needs to have only few contacts – it can forward its queries to these contacts – to obtain access to the whole network. This kind of free riding behavior has not received much attention so far. We refer to it as *deliberately poor connectedness (dpc)* or *contact-level free riding*. We refer to 'conventional free riding' where a peer does not process queries issued by other peers as *query-level free riding*. We will show that, without incentive mechanisms against dpc, dpc is often rational.

Our objective is to design mechanisms against dpc that are effective. This is not trivial: One must deal with network formation, i.e. dpc, and interaction selection, i.e. query-level free riding, in combination. To deal with dpc peers and peers that do query-level free riding it is not sufficient to take only connection costs into account, but also the costs and benefit of message handling. This is necessary to explore the dependency between interaction selection and contact selection.

In this paper, we analyze the effects of dpc using a cost-based model to show the following: dpc peers are more successful than cooperative ones in many situations if there are no mechanisms against dpc. Our model tells us when exactly this is the case. We have found out that the better peers differentiate between cooperative and uncooperative peers at the query level, the more advantageous is dpc. This is surprising: One might expect that mechanisms against query-level free riding would not have any relationship to dpc. To overcome dpc, we propose a new mechanism, the *C4C Mechanism*, that stimulates peers to establish additional contacts. The idea behind C4C is that peers without additional contacts cannot make use of additional contacts of other peers: a peer forwards a query only as far as the predecessor in the forwarding chain has done. The effect is that queries issued by dpc peers have a longer path length and are more likely to get lost than queries from cooperative peers. Thus, dpc peers have a higher ratio of unanswered queries and lower payoffs. The C4C Mechanism does not need extra messages or complex computations. It does not rely on any kind of statistics and is robust against churn.

Paper outline: We discuss related work in Section 2 and describe one specific structured P2P system in Section 3. We introduce our cost model (Section 4), describe different strategies (Section 5), and show that dpc dominates in many networks (Section 6). Section 7 proposes the C4C

Mechanism, Section 8 features a formal analysis. We then evaluate our mechanism experimentally, discuss our results and conclude.

2 Related Work

We discuss related work on cooperation issues, followed by network formation. We then cover work dealing with both issues in combination.

There exist models that analyze free riding on the query-level [10, 7] and countermeasures [14, 6]. [10, 7] do not take network formation, i.e. dpc, into account. [14] proposes a tamper-proof incentive mechanism for truth-telling. It could be used to force peers to be honest about their interaction mode. However, it requires a centralized infrastructure and side payments. These are not realistic in P2P systems. [6] shows that peers can use feedback attached to 'regular' messages to identify uncooperative peers. Both [14, 6] do not deal with dpc.

Let us now turn to related work on contact selection: [8] analyzes the network-formation process of selfish nodes for different network structures regarding performance and resilience. They observed that a network is more robust if all peers have almost the same number of contacts: If many peers resort to free riding, there are some (few) highly connected peers which become bottlenecks. In [8], nodes could not choose their interaction mode. [16] proposes an incentive mechanism based on taxes and subsidies that influences the mode of cooperation and the number of contacts per peer. This model is not applicable here since the nature of a connection, i.e. neighbor or additional contact, is not taken into account, and we are not aware of any coordinator-free infrastructure for taxes. Other network-formation models exist, [13] gives a good overview. The models are not applicable here because they do not address free riding.

Some literature on both strategy selection and network formation exists. [11] shows that the cost of adding a contact affects the strategies played. [1] showed that the network structure influences the behavior of peers. These models cannot be used to describe query-level free riding as well as contact-level free riding and their dependencies. [3] analyzes network formation with unreliable links. They show that most of the time only 'super-connected networks' are efficient. This means that peers in a network with unreliable links should establish more connections than in a reliable one. Still [3] does not investigate whether a peer has an incentive to be connected well. Another analysis of the simultaneous selection of interaction strategies and of the contacts [4] shows that some efficient network structures will not form without this simultaneous choice. Thus, network formation and interaction selection should be investigated in tandem.

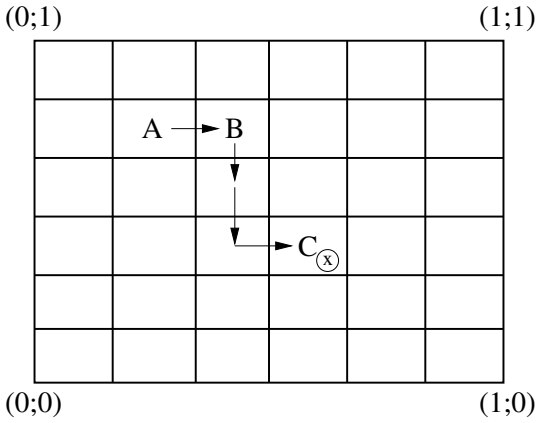

Figure 1. Two-dimensional CAN.

3 Content-Addressable Networks

Structured P2P systems manage sets of (key, value)-pairs. Content-Addressable Networks (CAN) are a prominent variant of such systems [17], and we will explain most of our points based on two-dimensional CAN. Still we believe that our results hold for other structured P2P systems as well, see Section 10.

To store data in a CAN, the keys of (key, value)-pairs are mapped to coordinates of a coordinate space, which is a d-dimensional torus. A peer administers all (key, value)-pairs mapped to its zone, i.e., a part of the coordinate space. Each peer also knows all peers with adjacent zones, its *neighbors*. A peer can query the value corresponding to a key, i.e., issue a query. To do so, it first transforms the key to coordinates, the *target of the query*. Queries are forwarded using greedy forwarding: The issuing peer calculates the distance of its neighbors to the target and sends the query to one with a small distance. This procedure recurs until the peer knowing the (key, value)-pair sought is reached. It then returns the value to the issuer.

Example: Figure 1 shows a 2-dimensional CAN. Rectangles represent the peer zones. Suppose that Peer p_A requests the value of key X, and the coordinates of X are in the zone of Peer p_C. Peer p_A does not know Peer p_C, but it knows that Peer p_B is closer to the target than itself. Peer p_B and other peers forward the query until it reaches Peer p_C. Peer p_C sends the query result to Peer p_A. ■

Additional Contacts. In a two-dimensional CAN, as described above, the cost of query evaluation is the number of hops, i.e., $\frac{\sqrt{n}}{2}$ on average, with n being the number of peers in the system [17]. [21] shows that the number of hops can go down significantly when using additional contacts for forwarding. [15] has investigated the routing complexity in random networks with an inverse power-law distribution,

i.e., connections to close contacts are favored over distant ones. It is proven that queries can always be delivered in $O(log^2(n))$ steps. This is significantly less than the CAN routing complexity. The existence of such additional contacts has an important benefit from the perspective of all peers: The number of peers that process a query and hence might drop or loose it is smaller.

In this paper, peers can become additional contacts of each other if they both agree. A peer which deems an additional contact uncooperative drops the connection. This is possible at any point in time.

4 Cost Model

In the presence of connection costs, peers will not form too many connections, and dpc is attractive. In the following we will show when exactly dpc is beneficial. We will introduce a network model that takes network formation and cooperation into account.

4.1 Network Properties and Assumptions

Each peer p_i has v^i initial contacts, its neighbors. It can add additional contacts or remove them. (It cannot remove its neighbors.) Note that this is similar to social networks: A person has relatives, neighbors etc. he cannot choose, but he can choose his friends.

Our formal analysis will make certain assumptions:

Homogeneity. All peer zones have the same size. In reality, CAN almost meet this condition. Their zones differ only by a small limited factor [17]. Further, all peers have the same utility function z which we will define in the next subsection.

Time. Time is a discrete sequence of points.

Querying. All peers issue queries at the same rate, one per point of time. Queries are equally distributed over the coordinate space. This assumption is realistic – using a hash function to map application keys to coordinates typically ensures this.

Additional Contacts. Maintaining a connection incurs costs proportional to its duration in time.

Lost Queries. There always are peers that do not forward or answer queries. We believe this is a realistic assumption: Even if all peers cooperate, some queries get lost due to technical failures.

Insertions; Tampering. This analysis leaves aside insertion of data objects into the CAN as well as tampering with data objects. While these are important topics, they exceed the scope of this paper.

Semantics	Variable	Utility Factor
number of received answers	α	a
number of queries issued	ρ	q
number of forwarded queries	θ	f
number of answers produced	ϕ	w
number of additional contacts	γ	c
number of points in time	r	

Table 1. Utility function – variables.

4.2 Utility Function

A structured P2P system consists of a set of peers $P = \{p_1, ..., p_n\}$. We assume that all peers are rational, i.e., they try to maximize their payoff. Each peer gains positive utility by obtaining results for queries it has issued. The gain of a query result is a, and a Peer p_i receives α_i query results during its lifetime. The revenue of Peer p_i is $revenue(i) = a \cdot \alpha_i$. To obtain a query result, a peer has to issue the query first. Issuing a query incurs the negative utility q. ρ_i is the number of queries issued by Peer p_i. A peer can forward a query at cost f and answer a query at cost w. The number of queries Peer p_i forwards in its lifetime is θ_i, and the number of queries that it answers is ϕ_i. As mentioned before, each peer can have additional contacts. Peer p_i must take costs c for every additional contact per point of time. γ_i is the average number of additional contacts over the lifetime of Peer p_i in the network. r_i is the time interval (i.e., number of points of time) Peer p_i is in the network (see Section 4.1). Table 1 is a summary of the abbreviations.

Thus, in its lifetime, Peer p_i must take the costs $costs(i) = q \cdot \rho_i + f \cdot \theta_i + w \cdot \phi_i + c \cdot \gamma_i \cdot r^i$. The utility function of p_i is $z(i) = revenue(i) - costs(i)$. Thus,

$$z(i) = a \cdot \alpha_i - (q \cdot \rho_i + f \cdot \theta_i + w \cdot \phi_i + c \cdot \gamma_i \cdot r_i) \qquad (1)$$

The utility function takes both network-formation costs and costs of queries sent to contacts into account. This facilitates simultaneous investigation of network formation and interaction selection. Like other approaches, e.g., [3, 4, 5, 7, 8, 9], we do not take the costs of simple computations, e.g., calculating a distance, into account. They are negligible, compared to the costs of query processing.

If we do not have a specific peer in mind, we omit subscript i. We denote the average over all peers with the $\bar{\cdot}$ operator. E.g., $\bar{\theta}$ is the average number of forward operations per peer.

5 Strategies

In an ideal system each peer should process any query it receives. In reality, peers use *cut-off strategies* [19], i.e.,

if a peer observes that a certain share of queries forwarded to a neighbor gets lost, the peer stops processing queries obtained from the neighbor.

By definition, *a peer is cooperative* if (a) it processes all queries received from its contacts, unless it deems the contact uncooperative, and (b) it establishes at least $\mathscr{J}$ connections to additional contacts and cuts connections to additional contacts that it deems uncooperative. ($\mathscr{J}$ is an exogenous, network-specific parameter; cf. [3].) *A peer is uncooperative* if it drops queries it receives from contacts that it deems cooperative. u_f is the ratio of queries not forwarded and u_a the ratio of queries not answered, both due to uncooperativeness. *A peer is deliberately poor connected (dpc)* if it establishes $\mathscr{J}_d < \mathscr{J}$ connections to additional contacts. Note that dpc is orthogonal to uncooperativeness: An uncooperative peer can or cannot be dpc.

dpc is attractive from the local perspective of a peer: The average path length of the queries it itself has issued increases by at most one hop, while its connection cost are low or zero. The peer can still issue queries using its neighbors. From a global perspective, however, the problem is that the peer does not help to lower the total forwarding load.

To keep our model simple, we assume that dpc peers have no additional contacts, i.e. $\mathscr{J}_d = 0$. Another reason for this simplification is that we want to investigate pure strategies first, before analyzing mixed ones.

Let p be the percentage of peers using a certain strategy. Superscript 'c' stands for a cooperative strategy. I.e., p^c is the share of cooperative peers in the network. Superscript 'd' stands for dpc peers (which are cooperative on the query-level), 'u' for uncooperative peers and 'ud' uncooperative dpc peers. The following condition must hold: $p^c + p^d + p^u + p^{ud} = 1$.

6 Formal Analysis of DPC

To understand in which situations dpc peers dominate cooperative peers, we derive the expected utility of dpc peers and cooperative peers.

Average Path Length. The expected utility of dpc peers and cooperative peers depends on the average path length. This is the average number of hops until a query reaches its target. More specifically, D_{max} is the average path length if no queries were dropped, and D_{avg} is the actual average path length including queries that have been dropped. Clearly, $D_{avg} \leq D_{max}$. The average path length changes if peers have additional contacts: Given a fixed number of queries, the more contacts peers have, the fewer queries they have to forward in total. In CAN without additional contacts, the average path length is $\frac{\sqrt{n}}{2}$ [17]. When the additional contacts of cooperative peers are distributed uniformly, the number of forwarding steps D_{max}^{coop} is

$$D_{max}^{coop} = \frac{1}{2}\int_0^{\frac{\sqrt{n}}{2}} \left(\omega_0(r)\cdot r + \omega_1(r)\cdot(r+\frac{\sqrt{n}}{2})\right) dr \quad (2)$$

with

$$\omega_0(r) = 4\cdot r\cdot\frac{\gamma+1}{n}\cdot e^{-2\cdot r^2\cdot\frac{\gamma+1}{n}} \quad (3)$$

and

$$\omega_1(t) = (-4t+2\sqrt{n})\frac{\gamma+1}{n}[1-\omega_0(\frac{\sqrt{n}}{2})]e^{(2t^2-2\sqrt{n}t)\frac{\gamma+1}{n}} \quad (4)$$

See [12] for a proof. Note that Formula 2 is a worst case estimation. Even if the second peer in the forwarding chain is dpc already, Formula 2 holds. Obviously, the following inequation holds:

$$D_{avg} \leq D_{max} \leq \begin{cases} D_{max}^{coop} & \text{with additional contacts,} \\ \frac{\sqrt{n}}{2} & \text{otherwise.} \end{cases} \quad (5)$$

Clearly, the average path length is related to the number of queries each peer has to forward. Given the average number of queries issued per peer $\bar{\rho}$, the average number of forwards per peer is the number of queries ($\bar{\rho}$) times their average path length:

$$\bar{\theta} = \frac{\bar{\rho}\cdot(D_{avg}-1)\cdot n}{n} = \bar{\rho}\cdot(D_{avg}-1) \quad (6)$$

In other words, for every query that has been issued, there are $(D_{avg}-1)$ forwarding steps. Subtracting 1 is because the peer which has issued the query does the first forward. (In our model the costs of the first forward are included in the costs of issuing a query.)

Influence of Uncooperative Peers. Next to the average path length, the overall effectiveness of the system depends on the ratio of uncooperative peers: Queries get lost when being forwarded to uncooperative peers. The probability that queries get lost grows with the ratio of uncooperative peers. It decreases with the ability to distinguish between cooperative and uncooperative peers. We abstract from the concrete classification technique that peers use and use an oracle with error probabilities f_p and f_n. These probabilities are constant values and are the same for all peers. f_p is the rate of false positives of the oracle, f_n the rate of false negatives. In other words, the oracle classifies an uncooperative peer as cooperative in f_p percent of the cases, and a cooperative one as uncooperative in f_n percent of the cases. A peer which wants to forward a query (or a result) to Peer p_c first asks the oracle whether p_c is cooperative or not. If so, it performs the operation. Otherwise, it chooses a new

contact. If no contact is cooperative according to the oracle, the peer drops the query.

The probability $\mathscr{F}$ that a query is forwarded depends on the probability that (a) it will not be forwarded to an uncooperative peer, or that the peer does not drop the query due to uncooperativeness ($\mathscr{F}_C := (p^c + p^d) + f_p(p^u + p^{ud})(1 - u_f)$) and (b) that the peer that receives the query does not deem the forwarder uncooperative ($\mathscr{F}_{DC} := (1 - f_n) + f_p$). It follows that $\mathscr{F} = \mathscr{F}_C \cdot \mathscr{F}_{DC}$. If the classification is perfect, i.e. f_p, f_n is close to zero, uncooperative behavior is not dominant, and $\mathscr{F}$ is close to one.

The probability $\mathscr{A}$ that the query is answered is analogous, except that the last forwarder cannot choose from its contacts; the peer that knows the query result is fixed: $\mathscr{A} = ((p^c + p^d) + (p^u + p^{ud})(1 - u_a)) \cdot \mathscr{F}_{DC}$.

Since a query has to be forwarded over $(D_{max} - 1)$ steps, the probability $\mathscr{P}(D_{max}, \mathscr{F}, \mathscr{A})$ that a query is processed successfully is:

$$\mathscr{P}(D_{max}, \mathscr{F}, \mathscr{A}) := \mathscr{F}^{D_{max}-1} \cdot \mathscr{A} \tag{7}$$

In other words, only some of the queries issued by a peer will be answered. The average number $\bar{\alpha}$ of queries answered is as follows:

$$\bar{\alpha} = \mathscr{P}(D_{max}, \mathscr{F}, \mathscr{A}) \cdot \bar{\rho} = \mathscr{F}^{D_{max}-1} \cdot \mathscr{A} \cdot \bar{\rho} \tag{8}$$

Utility of DPC. Inserting Formulae 6 and 8 into Formula 1 yields the expected utility of a cooperative peer, denoted as z_c:

$$\begin{aligned} z_c = &\mathscr{P}(D_{max}, \mathscr{F}, \mathscr{A}) \cdot a \cdot \bar{\rho} - (q \cdot \bar{\rho} + \mathscr{P}(D_{avg}, \mathscr{F}, \mathscr{A}) \\ &\cdot (D_{avg} - 1) \cdot f \cdot \bar{\rho} + w \cdot \phi + c \cdot \bar{\gamma} \cdot r) \end{aligned} \tag{9}$$

A dpc peer has a higher utility, for two reasons: First, it does not have to take any connection costs. Second, it receives fewer queries to forward, as it does not receive any queries from additional contacts. This means that dpc peers only forward $\frac{\nu}{\bar{\gamma}+\nu} \cdot \bar{\theta}$ queries.

Example: Consider a highly connected network. Let cooperative peers have four neighbors and 20 additional contacts on average, whereas Peer p_d has four neighbors and no additional contacts. Peer p_d only has to forward about $\frac{4}{24} = \frac{1}{6}$ of the queries a cooperative peer has to forward. ■

dpc however has a small drawback as well: The rate of queries issued by a dpc peer answered by the system is slightly less than the one of cooperative peers. But the issue is not severe: If a dpc peer has a cooperative neighbor, its queries only go through one additional hop. The probability $\mathscr{P}_d(D_{max}, \mathscr{F}, \mathscr{A})$ that a query issued by a dpc peer is answered is

$$\mathscr{P}_d(D_{max}, \mathscr{F}, \mathscr{A}) := \mathscr{P}(D_{max} + 1, \mathscr{F}, \mathscr{A}) \tag{10}$$

Hence, the expected utility z_d of a dpc peer is:

$$\begin{aligned} z_d = &\mathscr{P}_d(D_{max}, \mathscr{F}, \mathscr{A}) \cdot a \cdot \bar{\rho} - (q \cdot \bar{\rho} + \\ &\frac{\nu}{\bar{\gamma}+\nu} \cdot \mathscr{P}(D_{max}, \mathscr{F}, \mathscr{A}) \cdot (D_{avg} - 1) \cdot f \cdot \bar{\rho} + w \cdot \bar{\phi}) \end{aligned} \tag{11}$$

To see which strategy is dominant we now compare the utility of a cooperative peer to the one of a dpc peer.

Dominance of DPC. For some applications, the gain a of a query result may be small. Thus, we want to identify settings where dpc does not dominate cooperativeness even if the gain a of a query result is only slightly larger than the cost of forwarding a query (f) or of maintaining an additional contact (c).

Comparing the utility estimates from the previous subsection yields the following lemma:

Lemma 6.1 *Let the probability that a query is forwarded be less than 100%, i.e. $\mathscr{F} < 1$. dpc dominates cooperative behavior if the following inequation holds:*

$$a < \frac{\frac{\bar{\gamma}}{\bar{\gamma}-\nu} \cdot (D_{avg} - 1)}{(1 - \mathscr{F})} \cdot f + \frac{1}{(1 - \mathscr{F})} \cdot c \cdot \bar{\gamma} \tag{12}$$

See [12] for a proof. This lemma has two important implications: First, if the cost of forwarding a query (f) or of maintaining an additional contact (c) are high, cooperativeness is dominant only if the gain of a query result (a) is high. Second, if the probability $\mathscr{F}$ that a query is forwarded is high, dpc dominates cooperative behavior. Note that this is the case in particular if the classification is effective.

Example: Let there be 100,000 peers. 47.5% of these are cooperative, 50% are dpc, and 5% do not answer any queries ($p^c = 0.475, p^d = 0.475, p^u = 0.025, p^{ud} = 0.025$). Non dpc peers have 20 additional contacts ($\bar{\gamma} = 20$). Further, the cost of maintaining an additional contact (c) and of forwarding a query (f) are 1. Let the rate of false positives be 0.03 and the one of false negatives be 0.05. Lemma 6.1 implies that the gain of a query result (a) must be about 1085 times greater than the cost of forwarding a query (f) and the cost of an additional contact (c) so that dpc does not dominate cooperativeness. ■

If dpc is dominant, networks without an additional contact structure result, i.e., the CAN structure in our case. But as mentioned, more efficient structures exist – in settings where peers are not autonomous, and dpc is ruled out. We wonder if we can arrive at more efficient structures in our setting as well.

7 C4C – a Mechanism against DPC

In this section, we propose a cost-neutral mechanism against dpc, the C4C Mechanism. ('C' stands for contact.)

The idea is that peers without additional contacts cannot make use of additional contacts of other peers: A peer forwards a query only as far as the predecessor in the forwarding chain has done. In other words, cooperative peers reduce the utility of dpc peers: The probability that a query issued by a dpc peer is dropped increases, due to the higher path length. The mechanism gives dpc peers an incentive to change their behavior. It does not incur any extra costs due to extra messages, etc.

Algorithm 1: Forwarding of Queries

```
   Input: Query q, Peer p_s
1  Point t ← Target of query q;
2  Peer p_p ← The last peer which forwarded q to p_s;
3  Radius ρ ← Δ_{p_p,t} − Δ_{p_s,t};
4  if ρ < 0 then
5      return;
6  end
7  Array of Peers ℵ ← list of additional contacts of p_s;
8  foreach Contact p_c ∈ ℵ ordered by Δ_{p_c,t} ascending do
9      if Δ_{p_c,t} < ρ then
10         forward q to p_c;
11         return;
12     end
13 end
14 Forward q using closest neighbor of p_s that is
   cooperative;
```

By definition, $\Delta_{p_i,t}$ is the distance from the center of the zone of Peer p_i to Point t in the coordinate space. Algorithm 1 describes the C4C Mechanism: Before Peer p_s forwards a query, it computes the distance between the center of the zone of the predecessor p_p in the forwarding chain and the target t of the query. To determine how much the predecessor has shortened the forward distance, Peer p_s subtracts the distance $\Delta_{p_s,t}$ from $\Delta_{p_p,t}$ (Line 3). If the distance has increased, Peer p_s ignores the query (Lines 4-6). Peer p_s chooses a Contact p_c which is the closest contact to the target that is not further away than ρ from Peer p_s and forwards it the query (Lines 7-13). If such a peer does not exist, Peer p_s forwards the query to its cooperative neighbor that is closest to the target (Line 14). If there is no such neighbor, it drops the query.

Note that cooperative peers do not drop queries obtained from dpc peers completely. This would be problematic, because peers that have just entered the system would not have a chance of meeting other peers and establishing additional contacts.

By forwarding a query to a neighbor, dpc peers do not bring it much closer to its target. The neighbor in turn will only forward the query to its neighbor etc. It follows that the average path length of queries issued by dpc peers will be equal to the average path length in networks without an additional contact structure. In other words, there now is a direct relationship between contributing to the network structure and profiting from it. Establishing additional contacts becomes rational, as we will show in the next section. Further, our experiments will show that cooperative peers benefit more from the network than dpc peers. This is the case even though their queries are not routed via additional contacts as soon as they reach a dpc peer.

8 Formal Analysis of C4C

The formal analysis in this section lets us understand when C4C is effective. In the analysis and in our experiments we will assume that all cooperative peers use the C4C Mechanism. (Section 10 will discuss why this assumption is realistic.)

The C4C Mechanism ensures that peers without additional contacts cannot use the additional contacts of other peers. This means that dpc peers have the average path length of peers in a CAN without additional contacts whereas cooperative peers have a shorter path length (see Formula 5). The average path length for queries issued by dpc peers D_{max}^{dpc} is $\frac{\sqrt{n}}{2}$. I.e., $D_{max} = \frac{\sqrt{n}}{2} = D_{max}^{dpc}$ in Formula 10. However, the number of hops of queries issued by cooperative peers (D_{max}^{coop}) is much smaller. (See [12].)

Comparing the expected utilities yields the following:

Lemma 8.1 *Let the probability $\mathscr{F}$ that a query is forwarded be less than 100% ($\mathscr{F} < 1$). Then dpc dominates cooperation if the following equation holds.*

$$a < \frac{\frac{\bar{\gamma}}{\bar{\gamma}-\nu} \cdot (D_{avg} - 1)}{(1 - \mathscr{F} D_{max}^{dpc} - D_{max}^{coop})} \cdot f + \frac{1}{(1 - \mathscr{F} D_{max}^{dpc} - D_{max}^{coop})} \cdot c \cdot \bar{\gamma} \quad (13)$$

[12] contains a proof. Lemma 8.1 has one important implication, in contrast to Lemma 6.1, where dpc dominates cooperation without the C4C Mechanism: The more additional contacts cooperative peers have, the smaller becomes the average path length of queries from cooperative peers (D_{max}^{coop}), but not the path length of queries from dpc peers (D_{max}^{dpc}). Hence, the denominator in Formula 8.1 is about orders of magnitudes larger than without the mechanism. In large networks in particular, cooperative peers can break the dominance of dpc if they establish many connections. In these networks, effective classification does not necessarily lead to a dominance of dpc any more.

Example: Let all parameters be as in the previous example, except that cooperative peers use the C4C Mechanism. The gain of obtaining a query result must be greater than 19 so that cooperation dominates dpc. Compare this number to the setting without C4C: There, a must be greater than 1085.■

Lemma 8.1 describes the worst case for cooperative peers: Even if the issuer forwards its query only over one additional contact, Lemma 8.1 holds. If the query is forwarded over more than one additional contact, the results for cooperative peers should be even better.

9 Experiments

For our evaluation of the C4C Mechanism so far we have assumed that (a) all peers have zones of the same size, and that (b) dpc peers have no additional contacts at all. We will relax these assumptions in our experiments. The assignment of peers to zones is as in the original CAN paper [17]. dpc peers have between zero and five additional contacts (randomly chosen and uniformly distributed), and uncooperative peers drop between 50% and 100% of the queries on behalf of others; the exact rate is randomly chosen and uniformly distributed. The relaxation of (a) leads to a more natural network structure. Relaxing (b) reduces the effectiveness of our mechanism. Peers with few additional contacts can use some – but not all – of the additional contacts of cooperative peers, in spite of the C4C Mechanism. Still the objective of C4C is that these peers have a lower payoff than cooperative peers and a higher one than peers without any additional contacts.

To evaluate the influence of the distribution of dpc peers, the distribution of uncooperative peers, and the influence of the network size, we varied these parameters in three series of experiments. We measured the utility (z) of cooperative and dpc peers, to see which strategy is dominant. The figures also graph our predictions of the utility (z_c^+, z_d^+), to validate our formal analysis. For the prediction we assumed a rate of false positives of 0.025 and one of false negatives of 0.035. Our formal analysis had assumed that dpc peers do not have any additional contacts. Thus, we expect the predicted utility of dpc peers to be smaller than the measured one. In our experiments we have limited the number of additional contacts (γ) to 10. We have conducted experiments with other numbers, and the results were similar to the ones presented here. Part of our future work is to let peers themselves find good values for γ. All peers use a cut-off strategy to detect uncooperative peers. I.e., if a certain rate of queries sent from Peer p_i to its Contact p_c fails, p_i deems p_c uncooperative. This is in line with [19]: This work has shown that peers when controlled by humans use such strategies. The actual cut-off value is crucial: If it is too low, uncooperative peers are not recognized. If it is too high, cooperative peers are deemed uncooperative. In preliminary experiments, we had varied the cut-off threshold; 0.1% turned out to be good. In our experiments we used the following further parameters: The coordinate space has two dimensions, the gain of obtaining a query result (a) is 100, the cost of forwarding a message (f) and the one of an additional contact per point in time (c) are both 1. Issuing a query costs 2 ($q = 2$) and answering one costs 5 ($w = 5$). To avoid startup effects all measurements start after an initial phase of 500 rounds. Thus, the peers have formed a network structure already. (Investigating the dynamics of the system is future work.)

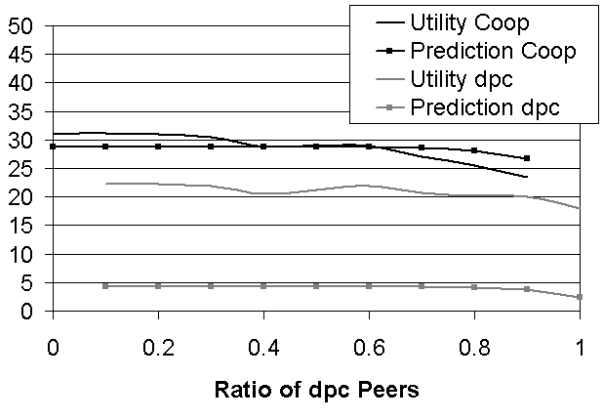

Figure 2. Influence of dpc peers.

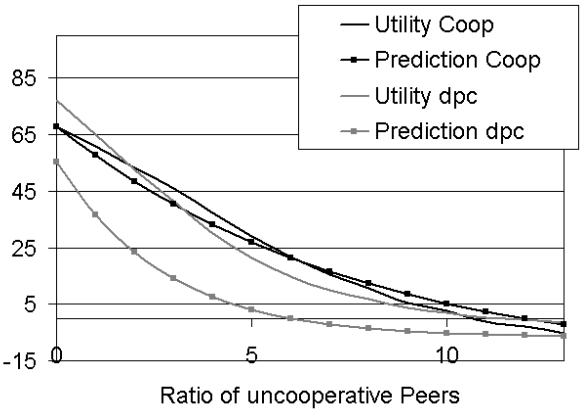

Figure 3. Influence of uncooperative peers.

With a first series of experiments, we test how the share of cooperative peers and dpc peers influences the effectiveness of our mechanism. The network consists of 5,000 peers. There are 5% uncooperative peers. Figure 2 graphs the result. The x-axis shows the percentage of dpc peers. The larger it becomes, the fewer non dpc peers are in the network. The y-axis is the utility per peer. First and foremost, the utility of cooperative peers using C4C is always higher than the one of dpc peers. dpc peers have less connection costs and slightly less forwarding costs, as long as there are cooperative peers in the network. On the other hand they have less income than cooperative peers if the C4C Mechanism is used.

In the second series of experiments, we varied the per-

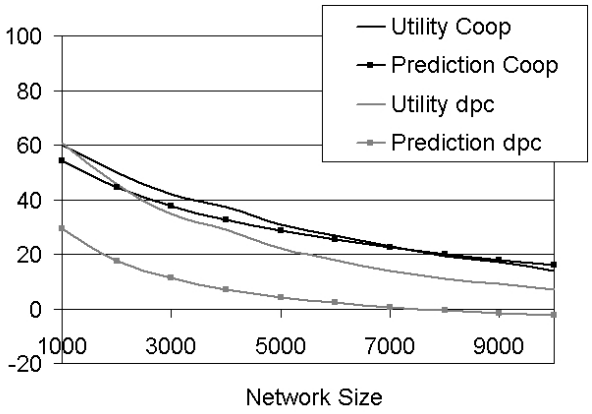

Figure 4. Influence of network size.

centage of uncooperative peers in the network. The remaining peers were either cooperative or dpc with the same probability. The network consists of 5,000 peers. Figure 3 shows the result: If the share of uncooperative peers is under 1%, dpc peers dominate cooperative peers. For higher ratios of uncooperative peers cooperation leads to a higher utility. As soon as more than 11% of the peers are uncooperative, cooperative peers and dpc peers do not benefit from the network any more, since too many queries get lost. It is not rational to participate in the system any more.

The predicted utilities and the ones observed in the experiments differ from each other, in particular when the rate of uncooperative peers is high. Our explanation (backed by further experiments not described here) is as follows: Our formal analysis uses global, fixed values for the rate of false positives and of false negatives of the classifier. In the experiment in turn, peers themselves estimate the degree of cooperativeness of their contacts. These estimations become more difficult with a higher rate of uncooperative peers, hence the difference in these settings.

Third, we investigated the influence of the network size. We varied it from 100 to 10,000 peers with 5% uncooperative peers, 47,5% cooperative ones and 47,5% dpc peers. Figure 4 shows the utility and the predicted utility for cooperative and dpc peers. Peers in larger networks have a lower benefit due to a larger average path length (and therefore a higher probability that a query is dropped). Again, dpc peers benefit less from the network.

All experiments are in line with our predictions. Our mechanism is robust against different distributions of cooperative, dpc and uncooperative peers as well as different network sizes. In our experiments, dpc dominates cooperation only in networks with few ($\leq 1\%$) uncooperative peers. This indicates that C4C is effective in many realistic settings. If cooperative peers want to break the dominance of dpc in networks with few ($\leq 1\%$) uncooperative peers, then, according to Lemma 8.1, they have to establish more connections.

10 Discussion

In the following we say why we think that our assumption that peers use the C4C Mechanism is realistic. We then discuss the generality of our results.

Tit-for-Tat. [2] has shown that Tit-for-Tat, i.e. behave like your contact has behaved, is the most effective strategy when actors play cooperation games. The C4C Mechanism leverages this result: Only a peer that has additional contacts profits from the additional contacts of another peer. Even though Tit-for-Tat strategies incur additional costs (for bookkeeping), participants still use them.

Limits of C4C. There are situations where C4C is not effective. This is the case when no query is lost ($\mathscr{F} = 1$). This in turn holds when there are no uncooperative peers, or the classification works perfectly. Further, there must not be any technical failures. These conditions in combination are rather unlikely.

Structured P2P Systems. dpc can occur with other structured P2P systems as well. Without the C4C Mechanism it is rational to drop connections to peers with a distant zone. A peer still has access to the network if cooperates with only those peers whose zones are adjacent or at least close to its zone.

Example: In Chord, peers form a ring [20]. A peer has neighbor contacts as well as distant ones. A dpc peer p_d would only establish connections to its right and left contact on the Chord ring. The fact that there is a specific topology that prescribes the distant contacts of a given peer does not make a difference: Peer p_d can drop all queries from the distant contacts. ■

Generalization. We believe that our findings are not only interesting for the P2P community, but for a broader audience. Any network with an underlying distance metric which consists of autonomous participants who can choose their contacts freely can benefit from our results, be it social or commercial. Participants can use the C4C Mechanism if they want other participants to establish additional contacts of their own.

11 Conclusions and Future Work

Networks should have a contact structure that facilitates routing with a low routing complexity. But since forming connections is costly, dpc, i.e., forming hardly any connections, is rational. In many situations, dpc dominates cooperation, as we have shown. Further, we have proposed an incentive mechanism against dpc, the C4C Mechanism, and have demonstrated its effectiveness. The idea is that

peers behave reciprocally on the contact-level, to stimulate dpc peers to establish more connections: A peer forwards a query only as far as the predecessor has done. This scheme is cost neutral and robust against changes of the network size or of the distribution of strategies in the network.

In the future, we will verify our theory by means of behavioral experiments, similar in spirit to [19].

References

[1] G. Abramson and M. Kuperman. Social games in a social network. *Physical Review E*, 63(3), Mar. 2001.

[2] R. Axelrod. *The Evolution of Cooperation*. Basic Books, September 1985.

[3] V. Bala and S. Goyal. A Strategic Analysis of Network Reliability. *Review of Economic Design*, 5(3), 2000.

[4] S. K. Berninghaus et al. Network formation and coordination games. *Advances in Understanding Strategic Behavior: Game Theory, Experiments, and Bounded Rationality*, 2004.

[5] V. Bhaskar and F. Vega-Redondo. Migration and the evolution of conventions. *Journal of Economic Behavior & Organization*, 55(3), 2004.

[6] E. Buchmann et al. Fairnet - How to Counter Free Riding in Peer-to-Peer Data Structures. In *CoopIS/DOA/ODBASE (1)*, 2004.

[7] N. Christin et al. A cost-based analysis of overlay routing geometries. In *IEEE INFOCOM'05*, 2005.

[8] B.-G. Chun et al. Characterizing selfishly constructed overlay networks. In *IEEE INFOCOM'04*, 2004.

[9] J. C. Ely. Local conventions. *Advances in Theoretical Economics*, 2(1), 2002.

[10] Z. Ge et al. Modeling peer-peer file sharing systems. In *IEEE INFOCOM'03*, 2003.

[11] S. Goyal et al. Learning, Network Formation and Coordination. *Games and Economic Behavior*, 50, 2005.

[12] B. Hartmann, K. Böhm, A. Khachatryan, and S. Schosser. The Dangers of Poorly Connected Peers in Structured P2P Networks and a Solution Based on Incentives. Technical Report 2007-16, Universität Karlsruhe, 2007.

[13] M. Jackson. A Survey of Models of Network Formation: Stability and Efficiency. *Group Formation in Economics: Networks, Clubs, and Coalitions*, 2003.

[14] R. Jurca et al. "CONFESS" . An Incentive Compatible Reputation Mechanism for the Online Hotel Booking Industry. In *IEEE Conference on E-Commerce*, 2004.

[15] J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000.

[16] H. Lugo and R. Jiménez. Incentives to cooperate in network formation. *Comput. Econ.*, 28(1):15–27, 2006.

[17] S. Ratnasamy et al. A Scalable Content-Addressable Network. In *SIGCOMM '01*, volume 31. ACM Press, 2001.

[18] S. Saroiu et al. A Measurement Study of Peer-to-Peer File Sharing Systems. In *SPIE MMCN*, 2002.

[19] S. Schosser et al. Incentives Engineering for Structured P2P Systems - a Feasibility Demonstration using Economic Experiments. In *ACM Electronic Commerce*, 2006.

[20] I. Stoica et al. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *ACM SIGCOMM*, 2001.

[21] Z. Xu et al. Building Low-maintenance Expressways for P2P Systems. Technical Report HPL-2002-41, HP Laboratories, 2002.