Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 29 (2006) 3567–3579
computer communications
www.elsevier.com/locate/comcom

Constructing efficient peer-to-peer overlay topologies by adaptive connection establishment ☆

Huirong Tian *, Shihong Zou, Wendong Wang, Shiduan Cheng

State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, PR China

Received 31 August 2005; received in revised form 27 May 2006; accepted 31 May 2006
Available online 28 June 2006

Abstract

A Reciprocal Capacity based Adaptive Topology Protocol (RC-ATP) is proposed in this paper to construct efficient peer-to-peer networks. It is based on the rational belief that a peer is only willing to maintain connections with those which will benefit it in future. Reciprocal capacity is defined based on peers' capacity of providing services and of recommending service providers. As a result, reciprocal peers connect each other adequately. Therefore, the resulting topologies are more efficient and resilient than Adaptive Peer-to-Peer Topologies (APT). Furthermore, RC-ATP has the intrinsic incentive to active peers as they are more advantaged and important in the network than freeriders and malicious peers. Although the overhead of the topology adaptation is a bit higher, RC-ATP works more efficiently with less cost compared with the Adaptive Peer-to-Peer Topologies Protocol (APTP).

Keywords: Peer-to-peer networks; Adaptive topology; Reciprocal capacity; Incentive compatibility; Trust

1. Introduction

Peer-to-Peer (P2P) networks have many benefits over traditional client-server approaches to cooperative working, data sharing and large scale parallel computing. Unfortunately, the P2P service availability is affected by the topology of P2P networks and the heterogeneity of peers' capacity.

The P2P overlay topology is very important to the P2P network performance such as scalability, efficiency and resilience. The most popular unstructured P2P systems (e.g. Gnutella) are not highly scalable or efficient because of peers' random inter-connections [1]. Alternative structured P2P schemes [2,3] are efficient for locating files but don't support semantic queries and are lack of the adaptation to highly dynamic P2P environments. Many studies focus on adaptive P2P topologies to improve the performance of Gnutella like P2P networks. The work [4–6] makes peers change neighbors based on their capacity of processing messages to avoid being overloaded. Some work[7–9] allows peers to choose neighbors according to their physical locality to reduce the message transfer latency. There are also some mechanisms[10,11] enable peers to connect others with similar interests to decrease the flooding queries. All these mechanisms can improve the performance of P2P networks to a certain extent. However, they don't discuss the issues of malicious peers or freeriders which impact the P2P service availability seriously.

It is measured that nearly 70% of Gnutella users share no files [12]. In addition, there still exist a large amount of P2P services with unreliable qualities and malicious actions [12]. In fact, In P2P networks, it is mainly the subjective configurations of peers that cause the heterogeneity of their capacity, but not the underlying physical infrastructures[13]. Therefore, in order to improve the P2P service availability, the peers' voluntary operations must be taken into account as well as the incentive mechanism.

☆ This paper was supported by the National 973 project of China (No.2003CB314806, No.2006CB701306), the NSFC (No.90204003, No.60472067).

* Corresponding author. Present address: P.O. Box 79 XI TU CHENG RD #10, 100876 Beijing, China. Tel.: +8610 62282007; fax: +8610 62283412.
E-mail address: tianhr@bupt.edu.cn (H. Tian).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.05.016

In order to design scalable, efficient, and robust overlay topologies, the Adaptive Peer-to-Peer Topologies Protocol (APTP) [1] takes the issues of malicious peers or freeriders as inherent parts of the topology design. In [1], a peer directly connects to those from which it is most likely to download satisfactory content. It adds or removes neighbors based on its local trust and connection trust of them which are decided by its transaction history. As we know, local trust and connection trust are determined by peers' own behavior and their neighbors' behavior respectively. The connection trust would vary with neighbors change. However, APTP doesn't consider the difference between local trust and connection trust. Once a peer unwittingly serves as a conduit through which a malicious peer disseminates inauthentic files and is disconnected by its neighbor, it will have no chance to connect this neighbor again even if it has provided many authentic files to this neighbor. This hinders good peers with similar interests getting connected adequately, which motivates us to do this work.

In this paper, a Reciprocal Capacity based Adaptive Topology Protocol (RC-ATP) for P2P networks is proposed to construct efficient P2P networks. In RC-ATP, it is assumed that the peer is rational. Hence, a peer is only willing to maintain connections with those which will benefit it in future. In order to keep connections with such peers, it should serve them in return. Reciprocal capacity is defined based on peers' capacity of providing services and of recommending service providers. As a result, reciprocal peers connect each other adequately. In addition, a response selection mechanism is proposed to reduce the probability of trying to download files from malicious peers. Therefore, the resulting topologies are more efficient and resilient compared with Adaptive Peer-to-Peer Topologies (APT). Furthermore, RC-ATP has the intrinsic incentive to active peers as they are more advantaged and important in the network than freeriders and malicious peers. Although the overhead of the topology adaptation is a bit higher, RC-ATP works more efficiently with less cost compared with APTP.

The rest of this paper is organized as follows: Section 2 presents related work. Reciprocal Capacity is defined in Section 3. In Section 4 we propose the reciprocal capacity based adaptive topology protocol for P2P networks. The simulation and analysis of the resulting P2P topology is followed. In the final section the conclusion and the direction of future work are stated.

2. Related work

Other related studies have been proposed in [4–11,14]. In [14], a P2P topology evolving algorithm is provided based on peers' global trust. However, in large scale P2P networks, the feasibility and the necessity of establishing global trust for every peer are still doubtful. B.F. Coper et al. [4] present a scheme to solve peers' overload problems by allowing them to self-organize into a relatively efficient network. In [4] a peer only disconnects peers when it becomes overloaded, where the connections between peers can be search links (through which search messages are sent) or index links (through which index entries are sent). A similar scheme is proposed in [5] and [6], where one type of link (search link) is considered. A peer rates its neighbors' capacity of processing queries and independently computes a level of satisfaction according to its own capacity and its neighbors' capacity. Then it gathers more neighbors with high capacity to improve the satisfaction level until it decides that its current set of neighbors is sufficient to satisfy its capacity. The purpose of [4–6] is to adjust the load between peers and make it match to peers' capacity. Neither of these schemes addresses the issues of malicious peers or freeriders in P2P networks. Y. Liu et al. [7–9] have studied adaptive unstructured P2P topologies. However, all these work focuses on the mismatching between the P2P overlay network and the physical underlying network and make peers choose closer nodes as neighbors to improve the performance of P2P networks. They do not care about malicious peers or freeriders. K. Sripanidkulchai et al. [10] have pointed that interested-based shortcuts often resolve queries quickly and avoid a significant amount of flooding in Gnutella. A similar mechanism[11] has also been proposed to dynamically reconfigure the overlay network with local clusters where peers share similar interests. Although both [10] and [11] can make queries reach the peers that satisfy them more quickly with less flooding messages, they don't discuss whether these peers would give responses or provide unreliable services.

3. Reciprocal capacity

In order to clarify the idea easily, we take the P2P network as a file-sharing network, although the proposed adaptive P2P topology protocol can be adopted by many other P2P service environments.

We define *reciprocal capacity* as a peer's subjective belief that other peers will benefit it in future. In the P2P network, what one peer can do for others is either providing the file directly or forwarding the query message to enable it to be responded by other peers. As shown in Fig. 1, the query issued by peer i is propagated to peer l through its neighbor peer k. After getting the response from l, peer i tries to download the file from peer l. Although peer k can not provide the file, but it makes peer i be responded by peer l. So we believe peer k recommends peer l and call it the recommender of peer l (the file provider). Thus, the capacity of peers in P2P networks is classified into two

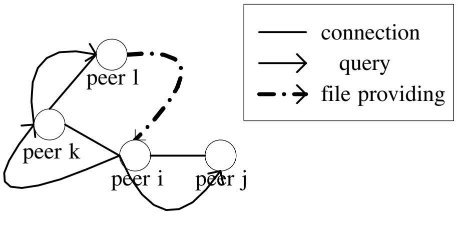


Fig. 1. Description model of capacities.

kinds: the capacity of providing services and the capacity of recommending service providers, denoted by *PSC* and *RSPC* respectively. Then the reciprocal capacity (noted as *RC*) is a weighted sum of these two kinds of capacities which are decided based on the peers' behavior history.

Before explaining how to get the value of *PSC* and *RSPC*, we define a property set $C^i = (\tau^i_{\min}, \tau^i_{\max}, w^i_{pro}, w^i_{rec}, f^i_{pnlt}, threshold^i_{cap}, win^i)$ for every peer i with constraints: $\tau^i_{\min} \leqslant \tau^i_{\max}$ and $w^i_{pro} + w^i_{rec} = 1$.

- $\tau^i_{\min}$ is the minimal connection number that peer i should maintain.
- $\tau^i_{\max}$ is the maximal connection number that peer i can serve.
- $w^i_{pro}(w^i_{rec})$ is the weight that *PSC* (*RSPC*) contributes to *RC* of peer j believed by peer i.
- f^i_{pnlt} is the penalty factor for peers' malicious actions in the current observing window. It is defined to prevent peers which have a good behavior history from concentrative malicious actions.
- $threshold^i_{cap}$ is the lowest reciprocal capacity that peer i can sustain with its neighbors. In other words, peer i will disconnect the neighbor whose reciprocal capacity is lower than $threshold^i_{cap}$.
- win^i is the observing window size.

It is believed that the peer's recent behavior has more impact on the future behavior in uncertain environments. So we define the observing window to distinguish the peer's recent behavior from the past long-term behavior.

It is assumed peer i has had transactions with peer j for M times. $Sat^p_{ij}(UnSat^p_{ij})$ is the number of satisfactory (unsatisfactory) transactions that peer i has had with peer j. We define PSC_{ij} as the probability that the $(M+1)th$ transaction with j will satisfy peer i. According to the deduction in [15], PSC_{ij} can be calculated by the following equation:

$$PSC_{ij} = \frac{Sat^p_{ij} + \alpha^p}{Sat^p_{ij} + UnSat^p_{ij} + \alpha^p + \beta^p}, \quad \alpha^p, \beta^p > 0, \tag{1}$$

where α^p and β^p are referred as hyper parameters, which represent the prior satisfactory and unsatisfactory transactions, respectively.

We introduce the observing window win^i and the penalty factor f^i_{pnlt}, and define *PSC* as follows:

$$PSC_{ij} = \frac{Sat^p_{ij} + \alpha^p}{Sat^p_{ij} + UnSat^p_{ij} + \alpha^p + \beta^p + \sum_{k=1}^{m^p_{ij}} f^i_{pnlt}}, \quad m^p_{ij} \geqslant 0, \tag{2}$$

where $UnSat^p_{ij}$ is redefined as the number of unsatisfactory transactions that peer i has had with peer j until the last observing window. m^p_{ij} represents the number of unsatisfactory transactions peer i has had with peer j in the current observing window. The factor f^i_{pnlt} is determined according to the unsatisfactory level.

Similarly,

$$RSPC_{ij} = \frac{Sat^r_{ij} + \alpha^r}{Sat^r_{ij} + UnSat^r_{ij} + \alpha^r + \beta^r + \sum_{k=1}^{m^r_{ij}} f^i_{pnlt}}, \quad m^r_{ij} \geqslant 0, \alpha^r, \beta^r > 0, \tag{3}$$

where Sat^r_{ij} is the number of satisfactory service providers that peer j has recommended to peer i (In the scenario of Fig. 1, if peer i is satisfied with the transaction with peer l, we say peer k has recommended a satisfactory service provider to peer i. Otherwise, peer k has recommended an unsatisfactory service provider to peer i), and $UnSat^r_{ij}$ is the number of unsatisfactory service providers that peer j has recommended to peer i until the last observing window. α^r and β^r are referred as hyper parameters, which represent the prior number of satisfactory and unsatisfactory service providers which have been recommended respectively. m^r_{ij} is the number of unsatisfactory service providers that peer j has recommended to peer i in the current observing window.

So the reciprocal capacity of peer j that peer i believes is defined as

$$RC_{ij} = w^i_{pro} \cdot PSC_{ij} + w^i_{rec} \cdot RSPC_{ij}. \tag{4}$$

When the system starts up, $Sat^p_{ij} = UnSat^p_{ij} = m^p_{ij} = 0$ and $Sat^r_{ij} = UnSat^r_{ij} = m^r_{ij} = 0$. So $PSC_{ij} = \alpha^p/(\alpha^p + \beta^p)$ (noted as PSC_{init}), which is the probability that a strange peer would provide a satisfactory service, and $RSPC_{ij} = \alpha^r/(\alpha^r + \beta^r)$ (noted as $RSPC_{init}$), which is the probability that a strange peer would recommend a satisfactory provider. Thus $RC_{ij} = w^i_{pro} \cdot PSC_{init} + w^i_{rec} \cdot RSPC_{init}$ (noted as *Init_Capacity*), which is the probability that a peer would get benefit from a strange peer.

In real P2P networks, peer i can dynamically configure its property set. The better the network connectivity is, the smaller $\tau^i_{\min}$ can be defined. $\tau^i_{\max}$ is determined by the capacity of processing incoming messages from neighbors. Generally, w^i_{pro} is larger than w^i_{rec} as the object of peer i entering the network is getting service. f^i_{pnlt} should be larger than 1 to punish peers' malicious behavior and be small enough to guarantee that good peers keep relative stable *RC* score if they unwittingly provide inauthentic files for uncontrolled reasons. $threshold^i_{cap}$ is usually larger than or equal to *Init_Capacity*. At the initial stage, win^i is defined relative small to make reciprocal peers get connected quickly. When the system is stable, win^i can be redefined a little large. However, if win^i is too large, peer i would not adjust neighbors duly and would not be sure of getting benefit from them and only serving reciprocal peers. α^p and β^p determine PSC_{init} which is strange peers' capacity of providing services. If it is measured that 70% of strange peers would provide authentic files, PSC_{init} would be 0.7 and α^p and β^p can be defined as 1.4 and 0.6, respectively when we set $\alpha^p + \beta^p = 2$. In order to reflect the peers' behavior timely, the sum of α^p and β^p would not be too large. If there is no measurement about PSC_{init}, optimistic peers may define it larger than 0.5 while pessimistic peers would define it smaller than 0.5. α^r and β^r can be defined similarly. For example, in our simulation, $\alpha^p,\beta^p,\alpha^r,\beta^r = 1$ is just based on the assumption that the probability of strange peers providing authentic files is the same as that of strange peers providing inauthentic files. we set $\tau^i_{\min} = 3$ and $\tau^i_{\max}$ is defined as 20 based on the measurement result in [19]. w^i_{pro} and w^i_{rec} is equal to 0.8 and 0.2, respectively just according to the eighty–twenty principle. f^i_{pnlt} is simply set as 2. We make $threshold^i_{cap}$ equal to 0.4 so that peers would not be disconnected just because they can't recommend good service providers when the system is at the initial stage. win^i is just as 1 to enable peers quickly find reciprocal peers.

4. RC-ATP

When a peer starts up, it uses the bootstrapping mechanism as that in Gnutella to find other nodes. Then, it directly sends a connection request to a random node until it has $\tau_{\min}$ neighbors. We assume it is not easy for peers to change identities so that the peers' behavior history will have the "shadow of the future" [16]. Peers initiate queries to locate services in P2P networks. These queries are broadcasted by the limited flooding-based method similar with that in Gnutella. In order to calculate *RSPC*, queries include an original neighbor field which is used to keep tracking of original neighbors to which peers initially send queries. The response appended with the original neighbor field of the query is directly sent to the query initiator.

In this section, the response selection mechanism is presented at first. The reciprocal capacity based topology adaptation mechanism is followed. Finally, we describe how to store and update the reciprocal capacity value of familiar peers.

4.1. Response selection mechanism

Peer *i*, upon receiving responses, will set a value for every response, noted as p_{gr}. This value reflects the probability that this response would bring a satisfactory transaction. If the responder *j* has had transactions with peer *i*, then $p_{gr} = PSC_{ij}$. Otherwise, $p_{gr} = RSPC_{ik}$, where *k* is the recommender of *j*.

Then, the responses, each corresponding to a responder which has had transactions with peer *i*, are ordered by p_{gr} descendingly if their p_{gr} is larger than PSC_{init}. This response list is called a good response list. The responses from strange peers whose p_{gr} is larger than $RSPC_{init}$, are also ordered by p_{gr} descendingly and appended to the good response list.

After obtaining the good response list, peer *i* will sequentially try to connect and download a file from the responders in the good response list. If it can not download a satisfactory file from the responders in the good response list, it will try the rest responses randomly until it get a good file or try every response.

How to choose query responses to download files is very important as it determines which peer to have a transaction with. If one peer maintains the connection with a neighbor with low *RSPC* and high *PSC*, there is a likelihood of getting malicious responses. This response selection mechanism can reduce the probability of trying to download files from malicious peers. In order to show the RC-ATP's effect against malicious peers, there isn't a response selection threshold to restrict the selection range in our response selection mechanism. However, as good peers only give responses when there is a match while malicious peers respond all queries, a response selection threshold is necessary to decrease the risk of the current transaction incurred by downloading files from malicious peers in real P2P networks.

4.2. Topology adaptation mechanism

It is believed that a success transaction can bring the P2P network profits. So we define the utility or productivity of the P2P network to be

$$P = \sum_{i=1}^{N_q} p_i(c_{qr}, c_{unsat}) - c, \tag{5}$$

where p_i, the profits that the *ith* query brings, is the function of c_{qr} and c_{unsat} (c_{qr} is the cost of processing query and response messages, and c_{unsat} is the cost of unsatisfactory transactions incurred by the *ith* query), c is the cost of topology maintenance, and N_q is the number of queries that peers have issued. As c_{qr} increasing, p_i decreases. And p_i decreases with c_{unsat} increasing. For example, we can define $p_i\ (c_{qr},\ c_{unsat}) = B - c_{qr} - c_{unsat} = B - \mu n_{qr} - \lambda B n_{unsat}$, where B is the utility of the success transaction corresponding to the *ith* query, μ is the CPU and bandwidth cost of processing one query or response message, n_{qr} is the number of query and response messages, λ represents the impact factor of unsatisfactory transactions, it can be defined according to the unsatisfactory level, λB is the cost of one unsatisfactory transaction for the *ith* query, and n_{unsat} is the number of unsatisfactory transactions incurred by the *ith* query. In file sharing P2P networks, if a peer issues a query, and downloads a satisfactory file of which the utility is 10 after trying three times. And for this query, 100 messages of queries and responses are handled in the network. When we define $\mu = 0.01$ and $\lambda = 0.1$, the profit that this query brings is $B - \mu n_{qr} - \lambda B n_{unsat} = 10–0.01 * 100–0.1 * 10*3 = 6$.

The cooperation of peers is of vital importance to the overall utility of P2P networks [19]. Intuitively, the more cooperative peers are, the more the network can produce. The reciprocal capacity can reflect the possibility that peers will cooperate in future. Therefore, we try to maximize *P* by a peer level protocol RC-ATP which makes every peer connect peers with large RC.

Topology adaptation happens when a observing window is finished. In order to get the profit, a peer will try to maintain $\tau_{\min}$ neighbors with reciprocal capacity larger than or equal to *Init_Capacity* and try to connect peers with large reciprocal capacity. Before introducing the topology adaptation mechanism, some notations are firstly defined as follows:

$Nb(i)$: the set of peer *i*'s neighbors.

$Fm(i)$: the set of the familiar peers of peer *i*, which are neighbors of peer *i* or have transactions with peer *i*.

$Fv(i) = \{j | RC_{ij} > Init\_Capacity, j \in Fm(i), j \notin Nb(i)\}$: the set of peers which peer *i* prefers to connect but hasn't connected yet.

$Fellow(i) = \{j | RC_{ij} \geqslant Init\_Capacity, j \in Nb(i)\}$: the set of peer *i*'s neighbors with reciprocal capacity larger than or equal to *Init_Capacity*.

$Fv(i)_{\max} = \{j | j \in Fv(i), \forall k \in Fv(i)$ and $k \neq j$, $RC_{ij} > RC_{ik}\}$: the peer belongs to $Fv(i)$ whose reciprocal capacity is the largest.

$Nb(i)_{\min} = \{j | j \in Nb(i), \forall k \in Nb(i)$ and $k \neq j$, $RC_{ij} < RC_{ik}\}$: the neighbor of peer *i* whose reciprocal capacity is the lowest.

4.2.1. Sending a connection request

When it's time to adapt the topology, peer *i* will do the following steps:

First, If $|Fellow(i)| < \tau^i_{\min}$ and $Fv(i) \neq \emptyset$, peer *i* will send a connection request to $Fv(i)_{\max}$, and remove it from $Fv(i)$; If $|Fellow(i)| < \tau^i_{\min}$ and $Fv(i) = \emptyset$ it will just send a connection request to a strange peer in the network randomly. If the connection request is accepted and $|Nb(i)| > \tau^i_{\max}$, $Nb(i)_{\min}$ will be disconnected.

Peer *i* will try to get connected with $\tau^i_{\min}$ neighbors whose reciprocal capacity is not lower than *Init_Capacity*. But if it tries several times and no peer accepts its connection requests, it has no choice but to give up.

Second, peer *i* examines neighbors' reciprocal capacity and disconnects the neighbor whose reciprocal capacity is lower than $threshold^i_{cap}$.

Third, peer *i* will send a connection request to $Fv(i)_{\max}$ and remove it from $Fv(i)$ when one of the following conditions is true: (1) $Fv(i) \neq \emptyset$ and $|Nb(i)| < \tau^i_{\max}$; (2), $Fv(i) \neq \emptyset$, $|Nb(i)| = \tau^i_{\max}$ and the reciprocal capacity of $Fv(i)_{\max}$ is larger than that of $Nb(i)_{\min}$. In the second case, $Nb(i)_{\min}$ will be disconnected if the connection request is accepted.

4.2.2. Receiving a connection request

Peer *j* will only accept the connection request from peer *i* if RC_{ji} is larger than *Init_Capacity* and one of the following conditions is true: (1) $|Nb(j)| < \tau^j_{\max}$; (2) RC_{ji} is larger than the reciprocal capacity of $Nb(j)_{\min}$. In the second case, peer *j* will disconnect $Nb(j)_{\min}$.

There are two issues not addressed by the above topology adaptation algorithm.

No-reciprocal peers. The issue of freeriders arising in P2P routing is that freeriders choose not to forward queries for others to conserve local bandwidth. Peer *i* may find that its neighbors neither provide files to it nor recommend file providers to it. This is the case that peer *i* enters the network from the wrong place or its neighbors are freeriders which only download files, and neither forward query messages nor share their own files. We define the notion of a no-reciprocal peer to be a peer whose reciprocal capacity is lower than or equal to *init_Capacity* (the no-reciprocal concept can be redefined according to the requirements of different P2P applications or different peers' opinions. For example, one peer may believe another peer, whose reciprocal capacity is lower than $Init\_Capacity + 0.1$ is no-reciprocal). Peer *i* will disconnect neighbors which are still no-reciprocal peers after several observing windows.

No responses. Although peer *i* disconnects its no-reciprocal neighbors, there is still an issue that it may get no responses for a period. The reason is that *i*'s neighbors may only forward query messages for *i* or provide *i* a few authentic files at the initial stage to increase their chances of maintaining connections with *i*. So when no responses are received for several observing windows, peer *i* will replace $Nb(i)_{\min}$ with a random strange peer for $\tau^i_{\min}$ times.

In RC-ATP, there is two-level topology adaptation. Adapting neighbors based on reciprocal capacity after a observing window can be treated as a short-term adaptation. It is a long-term adaptation that dropping no-reciprocal peers or replacing $Nb(i)_{\min}$ for no responses after several observing windows. Peers adjust neighbors in a short term for their interests in a quick profit while the long-term adaptation guarantee them get profit from neighbors and only serve reciprocal peers.

4.3. Maintaining the capacity value of familiar peers

4.3.1. Storing the capacity value of familiar peers

Every peer should keep a local data structure *history* = (Sat^p, $UnSat^p$, m^p, $Pnlt^p$, Sat^r, $UnSat^r$, m^r, $Pnlt^r$) for its every familiar peer, where the notations of Sat^p, $UnSat^p$, m^p, Sat^r, $UnSat^r$ and m^r are the same as those in Eqs. (2) and (3), $Pnlt^p$ represents $\sum_{k=1}^{m^p} f_{pnlt}$ in Eq. (2) and $Pnlt^r$ is equal to $\sum_{k=1}^{m^r} f_{pnlt}$ in Eq. (3).

Note that, m^p, $Pnlt^p$, m^r and $Pnlt^r$ maintain the data in the current observing window. One byte is allocated for each of these four parameters. Sat^p, $UnSat^p$, Sat^r and $UnSat^r$ are used to record the information of familiar peers all the life. Four bytes for each of them is enough. So the information of one familiar peer can be stored with 20 bytes. The storage cost of each peer even in the worst case is limited by the number of other peers having interactions with. In addition, we discard the stale information of peers periodically to save storage.

4.3.2. Updating the capacity value of familiar peers

If peer *i* is satisfied with the transaction with peer *j*, Sat^p_{ij} increases by one. If $j \notin Nb(i)$, there must be a peer $k \in Nb(i)$ which is the recommender of peer *j*. Thus Sat^r_{ik} is also increased by one. If peer *i* isn't satisfied with the transaction with peer *j*, m^p_{ij} increases by one and $Pnlt^p_{ij}$ increases by f^i_{pnlt} which is determined by the unsatisfactory level.

If $j \notin Nb(i)$ and peer $k \in Nb(i)$ is the recommender of peer j, m^r_{ik} increases by one and $Pnlt^r_{ik}$ increases by f^i_{pnlt}.

When the observing window ends and peer i has adjusted its connections, $UnSat^p_{ij}$ increases by m^p_{ij}, $UnSat^r_{ij}$ increases by m^r_{ij}, and $Pnlt^p_{ij}$, $Pnlt^r_{ij}$, m^p_{ij} and m^r_{ij} are set with 0 for peer $j \in Fm(i)$.

5. Simulation and analysis

RC-ATP is implemented based on Query Cycle Simulator [17,18] and is compared with APTP. The experiment data of APTP is obtained by running the demo of adaptive topologies in [17]. There are 800 query cycles in one experiment and the results are averaged over 5 runs.

5.1. Simulation environment

The simulation is based on a query cycle model in a file-sharing P2P network [18]. In each query cycle, a peer i in the network may be actively issuing a query, inactive, or even down and not responding to queries passing by. Upon issuing a query, a peer waits for incoming responses, selects a download source among those peers that have given responses and starts downloading the file until gets the authentic file or tries all the download sources. Then the query cycle finishes and the data is collected [18].

The network is initialized as the random graph where peers have $\tau_{\min}$ neighbors. The query message is flooded with TTL = 4. In the experiment, there are 500 normal peers and 50 malicious peers (providing inauthentic files in order to undermine the network performance). 25% of the normal peers are freeriders (only downloading files, and neither sharing their own files nor forwarding queries) while the others are good peers(normally downloading and uploading files). Let α^p, β^p, α^r, $\beta^r = 1$ so that *Init_Capacity* = 0.5. To simplify the scenario, all peers have the same property set $C = (3,20,0.8,0.2,2,0.4,1)$. Connection requests from peers whose local trust scores are −1 or connection trust scores are −5 will not be accepted in APTP. Normal peers are in the uptime with the uniform random distribution over [0%, 100%] and issue queries in the uptime with the uniform random distribution over [0%, 50%], while malicious peers are always up and always issue queries. In addition, different types of peers also vary in their behavior of responding queries and providing files. For good peers, the probability of providing inauthentic files is 5%, while malicious peers will respond to all queries they have received and provide inauthentic files for all download requests.

The content distribution model is the same as that in [18]. In this model each file is characterized by the content category c and the popularity rank r within this category. c and r both follow the Zipf distribution. Files are distributed probabilistically to peers based on their popularities and the content categories that peers are interested in. Distributions used in the model are based on the measurement of real-world P2P networks [19]. Normal peers issue queries in accordance with their interests while malicious peers issue queries randomly just to know good peers.

In our simulation environment, 20 content categories are hold in the network and each peer is at least interested in four categories.

5.2. Efficiency

The efficiency of the network describes how good peers can efficiently get reliable files. The metrics used to evaluate the efficiency are as follows:

- The Ratio of the Authentic Response (RAR): reflects the average ratio of responses which are given by good peers to the total responses.
- The Probability of the Success Download (PSD): reflects the average probability that the first download is an authentic file.
- The Ratio of the Success Query (RSQ): reflects the average ratio of the queries which are satisfied with authentic files to the total queries.
- The Minimal Hop of the Authentic Response (MHAR): reflects the average minimal hop within which an authentic file is located.

If good peer i has issued a query and received $r_i(>0)$ responses among which r^a_i are given by good peers, then $RAR_i = r^a_i / r_i$. If peer i downloads an authentic file after trying λ_i times, $PSD_i = 1/\lambda_i$ and $RSQ_i = 1$. Otherwise, $PSD_i = 0$ and $RSQ_i = 0$. For all these r^a_i good responses, the minimal hop between responders and i is $MHAR_i$. We define the RAR, PSD, RSQ and MHAR of the network separately as the average of RAR_i, PSD_i, RAR_i and $MHAR_i$ over all such good peers in the network.

Fig. 2 plots the ratio of the authentic response of RC-ATP and APTP. The RAR of APTP increases rapidly at the initial stage because of the adoption of the connection trust. However, in RC-ATP with the query cycle

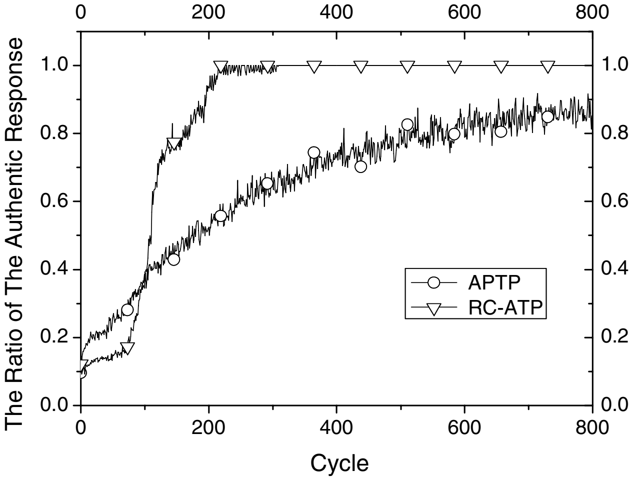


Fig. 2. The ratio of the authentic response.

increasing, malicious peers are quickly known by other peers and eliminated from the network. As a result, the RAR of RC-ATP will be larger than that of APTP at cycle 102 and be stable at 1 after 311 cycles.

The probability of the success download of RC-ATP and APTP is shown in Fig. 3. As the query cycle increasing, the PSD of RC-ATP is approximately 0.95, which is the ratio that good peers provide authentic files for download requests. Note that, at the initial stage PSD of RC-ATP is larger than that of APTP although RAR of RC-ATP is lower than that of APTP. This is because the response selection mechanism reduces the probability that good peers try to download files from malicious peers. The larger the PSD is, the lower the cost incurred by unsatisfactory transactions is and the larger the productivity of the network is.

In order to evaluate the effects on the network efficiency imposed by freeriders, we examine the ratio of the success query of RC-ATP and APTP in the condition of the freeriders fraction of normal peers *ffraction* = 0.25 and of *ffraction* = 0.75 separately. In Fig. 4 when *ffraction* = 0.25, the RSQ or APTP is about 70% as well as that of RC-ATP. When *ffraction* = 0.75, the RSQ of RC-ATP increases more rapidly than that of APTP because in RC-ATP dropping no-reciprocal peers makes good peers find reciprocal peers quickly. The RSQ is not equal to 1 because there are freeriders in the system and some good peers are inactive and do not handle the query. When RSQ is stable, it is larger in the condition of *ffraction* = 0.25 than that in the condition of *ffraction* = 0.75. As expected, the lower *ffraction* is, the more efficient the network is. In addition, we can also conclude from Fig. 4 that RC-ATP is more efficient than APTP when the freeriders fraction is large.

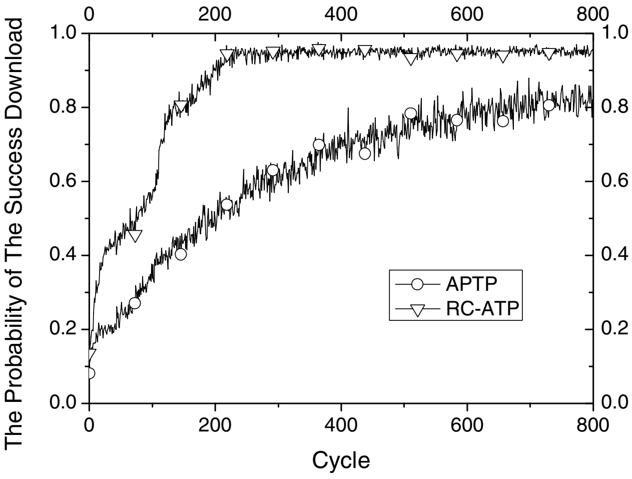


Fig. 3. The probability of the success download.

Fig. 5 compares the minimal hop of the authentic response of RC-ATP and APTP. That of RC-ATP has a little improvement compared with APTP.

The philosophy of RC-ATP is that connections between peers are made based on reciprocal capacity. Hence, gradually peers maintain connections with others that share their interests. Such a network can be described as a small world network, which has a short average shortest path length and a high cluster coefficient. The cluster coefficient can be used to evaluate how well the network forms a clique. For peer i with k_i neighbors, the cluster coefficient c_i is defined as $c_i = \frac{2E_i}{k_i \cdot (k_i - 1)}$ [22], where E_i is the number of edges actually existing among the k_i neighbors. The nodes can communicate with each other efficiently in the network with small word characteristics. We evaluate the average shortest path length between good peers (If there is no path between two good peers, the average shortest path length between them is defined as $pl_{\max} = 15$) and the average cluster coefficient of good peers in Fig. 6 and Fig. 7. The average shortest path length between good peers in RC-ATP is larger than and tends to be similar with that of APTP. The reason is that peers disconnect no-reciprocal peers after several observing windows in RC-ATP. Even so, the requested authentic file can be located within fewer hops in RC-ATP than in APTP. The average cluster coefficient of good peers in APTP is larger than that in RC-ATP.

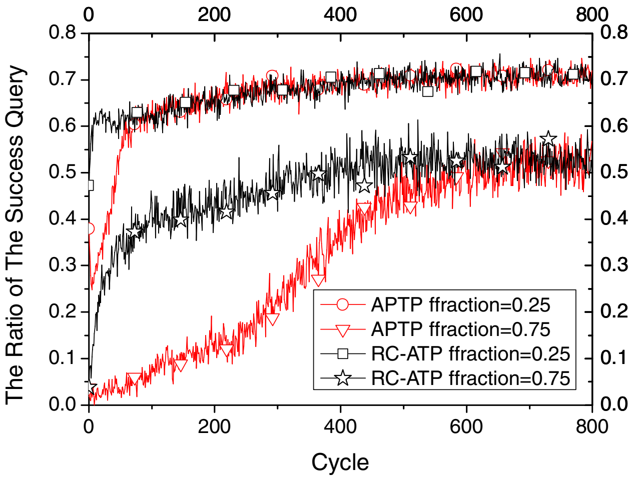


Fig. 4. The ratio of the success query.

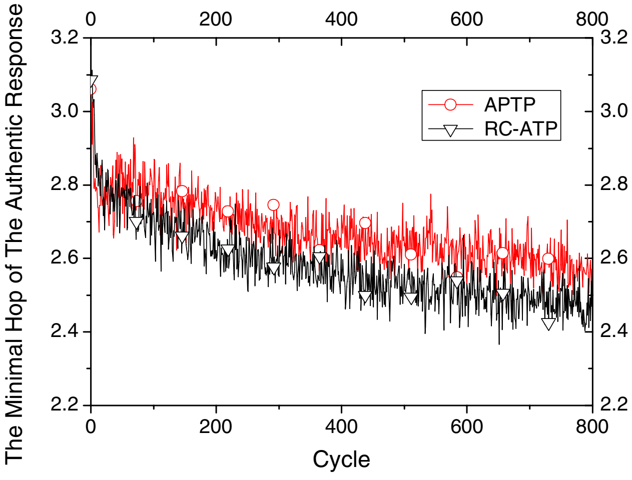


Fig. 5. The minimal hop of the authentic response.

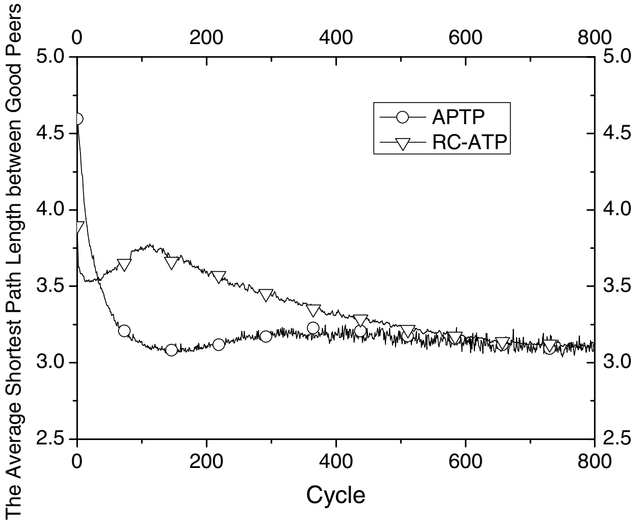


Fig. 6. The average shortest path length between good peers.

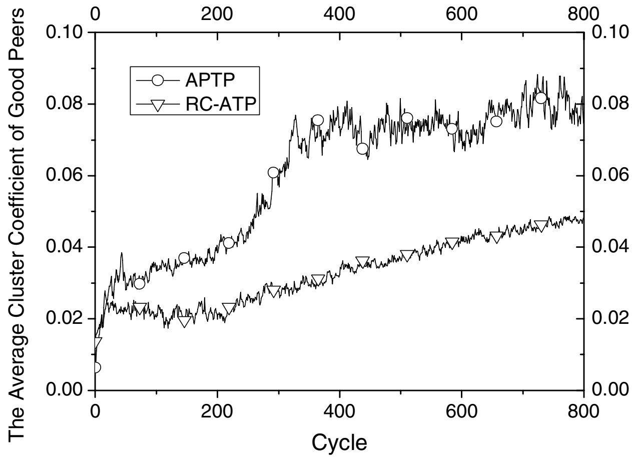


Fig. 7. The average cluster coefficient of good peers.

The more clustered peers are, the more traffic is incurred within the clique and the less chance there is for queries to be propagated to remote peers. However, in the large scale P2P network, it is very important to have wide horizon to get more information of the network.

From the above analysis, we can get the conclusions: the resulting network of RC-ATP is a small world network. Compared with APTP, good peers can more efficiently get reliable files in RC-ATP. In addition, RC-ATP has more advantage than APTP when the freeriders fraction is large.

5.3. Resilience

In general, the network resilience studies the effect of removing nodes from the network. Such attacks are classified into two categories: random failures (removing nodes randomly) and intentional attacks (removing nodes intentionally) [23]. In this paper, we evaluate the resilience of the resulting topology of RC-ATP and APTP under random failures (Failure) of nodes and two kinds of intentional attacks: degree-based attacks (denoted by DAttack) and betweenness-based attacks (BAttack). Three metrics are analyzed: the relative size of the largest cluster S (the ratio between the size of the largest cluster and the size of the network), the average shortest path length between good peers pl (the average shortest path length between the good peers remaining in the network) and the network diameter d.

Fig. 8 summarizes the results of Failure, DAttack and BAttack of nodes measured by S, pl and d at cycle 800 as functions of the fraction of removed nodes f in [0, 0.15](Nodes are selected from the original 550 nodes). In Fig. 8a the initial S of RC-ATP and APTP is not equal to 1 and the former is lower than the later. This is because in RC-ATP not only all malicious peers but also some freeriders lose connections with the good peers cluster while almost all malicious peers are eliminated from the network in APTP at cycle 800. In Fig. 8a,b and c S decreases, pl increases and d increases in APTP more rapidly than in RC-ATP under DAttack and BAttack of nodes. So the network connectivity and the reachability between good peers

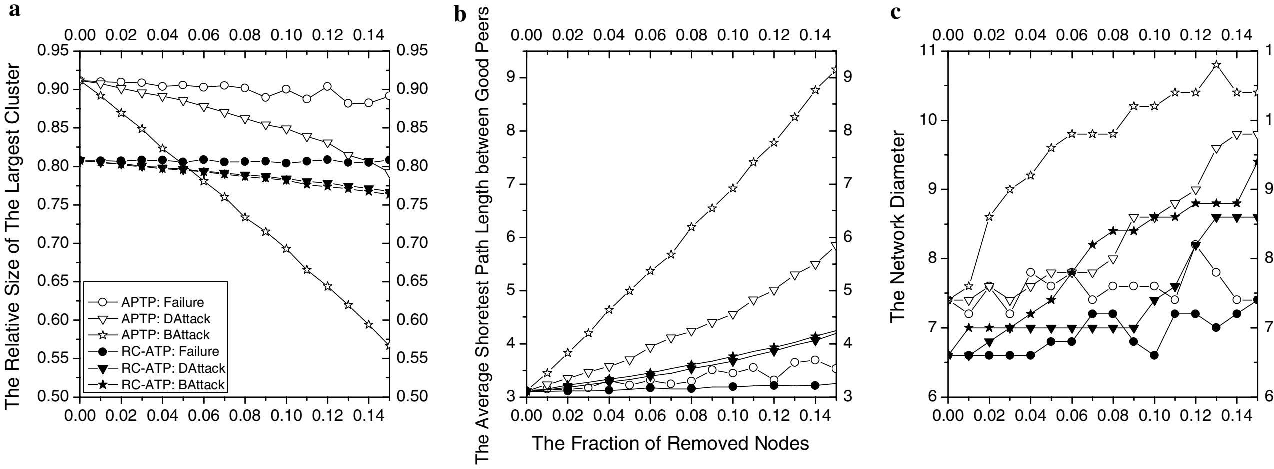


Fig. 8. Results of Failure, DAttack and BAttack of nodes measured by S, pl and d.

are better in RC-ATP than in APTP under intentional attacks. The network of RC-ATP is resilient not only to random failures but also to intentional attacks while the network of APTP is only resilient to random failures. The reason is that RC-ATP distinguishes the different capacities of peers and reciprocal peers are connected adequately. It can be concluded that the resulting topology of RC-ATP is more resilient than that of APTP. In addition, intentional attacks, especially BAttack of nodes degrade the functionality of the network more seriously than Failure.

5.4. The intrinsic incentive to good peers

The intrinsic incentive of the resulting topology to good peers is measured with the centralities.

The prominence of nodes in a network can be embodied by the centralities [20]. There are three centrality measurements: degree centrality, closeness centrality and betweenness centrality. Their formal definitions [21] are as follows:

- Degree centrality is a measure of the local centrality concerned with the relative prominence of a node in its neighbors. It is defined as $C_D(i) = D_i/(N-1)$, where $D_i = |Nb(i)|$ and N is the network size. The node with higher degree centrality will have a wider view of the network.
- Closeness centrality is defined as: $C_C(i) = 1/\sum_{j \in V} pl_{ij}$, where pl_{ij} is the shortest path length between node i and node j, and V is the set of the network nodes.
- Betweenness centrality is defined as $C_B(i) = \sum_{j \neq i \neq k \in V} \sigma_{jk}(i)/\sigma_{jk}$, where σ_{jk} is the number of the shortest paths between node j and k, and $\sigma_{jk}(i)$ is the number of the shortest paths between j and k which cross through node i.

Closeness centrality and betweenness centrality are the global centralities concerning the prominence of a node within the whole network. The node with higher closeness centrality will communicate with other nodes more quickly. So it is more advantaged than others in P2P networks. The node with higher betweenness centrality plays a central role in the network as it acts as an agent between the communications of other nodes. So the failure of such nodes will impact the performance of the whole network seriously (This is demonstrated by Section 5.3).

Fig. 9 shows the centralities of different types of peers. In Fig. 9a and c the degree centrality and the betweenness centrality of malicious peers are larger than that of freeriders at the initial stage. The reason is that malicious peers are always up, issue queries and forward queries for others while freeriders do nothing for others. So they have more chance to know other peers and get connected with them than freeriders at the initial stage. The degree centrality of malicious peers are lower than 10^{-4} after cycle 192 and stable with 0 after cycle 308 in Fig. 9a. And the betweenness centrality of malicious peers is equal to 0 after cycle 122 in Fig. 9c. This is because malicious peers are eventually driven to the fringe of the network and finally are eliminated from the network. In Fig. 9b and c, the closeness centrality and the betweenness centrality of good peers decrease rapidly as well as the closeness centrality of freeriders around cycle 600. It is because freeriders begin to be excluded from the network as no-reciprocal peers. The decrease of closeness centrality of good peers will not degrade the performance of them because freeriders don't forward queries for others although they contribute to the connectivity of the network. This is demonstrated by Figs. 4 and 5.

So the resulting topology has the intrinsic incentive to good peers as they are more advantaged and important in the network than freeriders and malicious peer. In order to demonstrate the effects of the topology incentive, we also evaluate other three metrics of good peers and freeriders:

- the Ratio of the Success Query (RSQ)
- the Minimal Hop of the Authentic Response (MHAR)
- the Ratio of the Topology Change (RTC) : for peer i, if it has changed connections c_i times in the observing window and gets $|Nb(i)|$ neighbors finally, $RTC_i = c_i/|Nb(i)|$. Thus, the RTC of good peers and freeriders are the average of RTC_i over good peers and freeriders respectively.

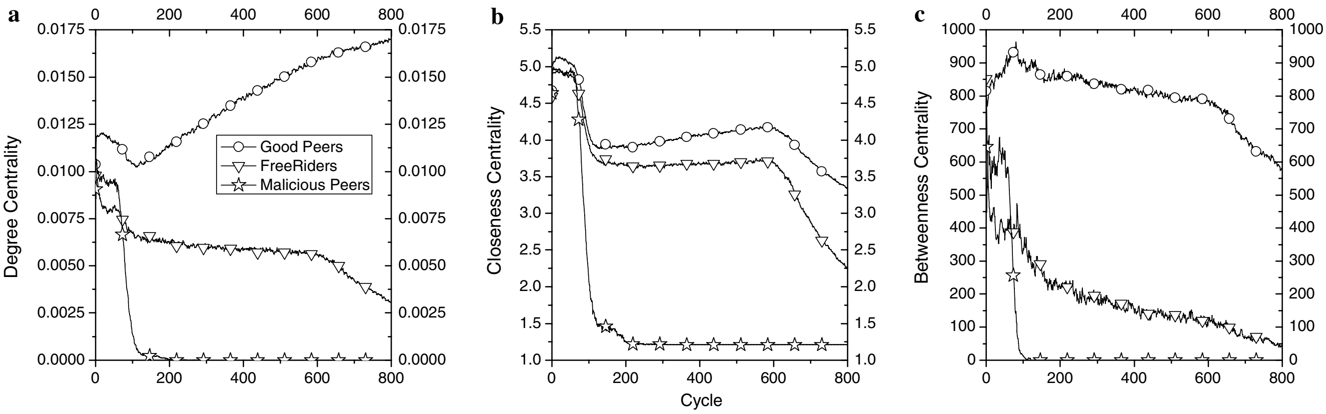


Fig. 9. The centralities of different types of peers.

These three metrics of good peers and freeriders are examined in the condition of *ffraction* = 0.25 and of *ffraction* = 0.75 respectively. In Fig. 10 when *ffraction* = 0.25, the RSQ of freeriders is close to that of good peers before cycle 610. And after cycle 610 the RSQ of freeriders decreases rapidly while that of good peers keeps stable. When *ffraction* = 0.75, the RSQ of good peers is always larger than that of freeriders. We also notice that the RSQ both of good peers and of freeriders in condition of *ffraction* = 0.25 is larger than that in condition of *ffraction* = 0.75. In Fig. 11 the MHAR of good peers is always lower than freeriders whether *ffraction* = 0.25 or *ffraction* = 0.75. Also we can get the MHAR of good peers is lower in condition of *ffraction* = 0.25 that in condition of *ffraction* = 0.75. In Fig. 12 the ratio of the topology change of freeriders reaches 0.5 while that of good peers is around 0.002 whether *ffraction* = 0.25 or *ffraction* = 0.75.

So, we can get the conclusions from Figs. 10–12 as follows: if *ffraction* is low, freeriders can locate the requested files as well as good peers by frequently trying to establish connections with strange good peers before they are recognized. But good peers locate the requested authentic files within fewer hops than freeriders. When *ffraction* is large, not only the performance of freeriders degrades but also the performance of good peers. In RC-ATP good peers can establish stable relationship with other good peers while freeriders have to frequently change their neighbors to get their profit.

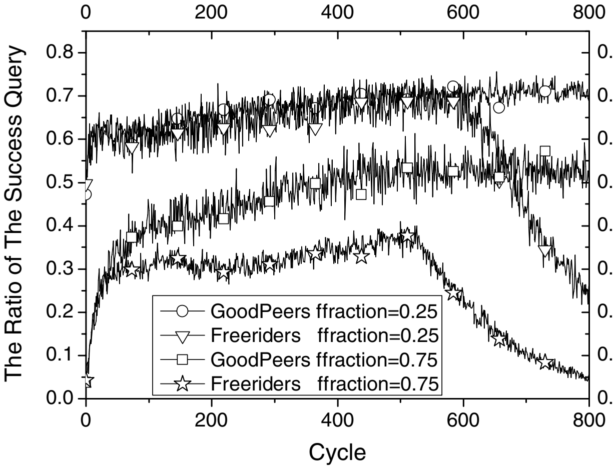


Fig. 10. The ratio of the success query of good peers and freeriders.

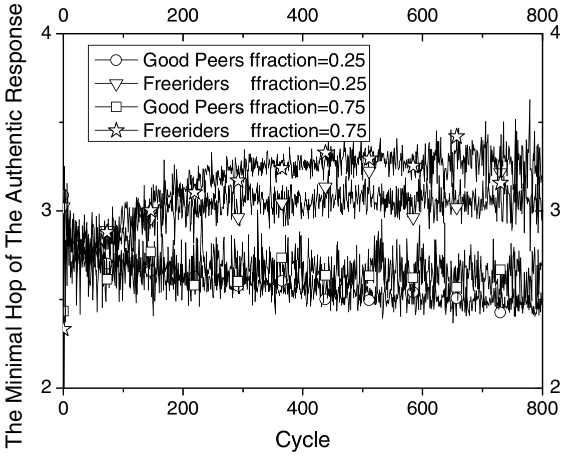


Fig. 11. The minimal hop of the authentic response of good peers and freeriders.

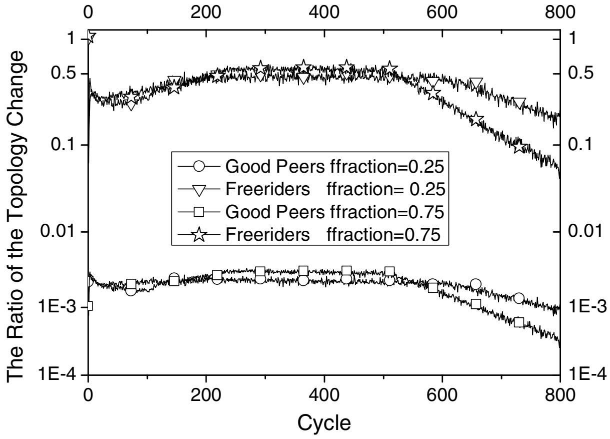


Fig. 12. The ratio of the topology change of good peers and freeriders.

In sum, it can be concluded that good peers in the resulting topology of RC-ATP have wider horizon of the network, can communicate with other good peers more efficiently and play a more central role than freeriders and malicious peers. As a result the performance of good peers is better than that of freeriders while malicious peers are eliminated from the network. Thus, the topology has the intrinsic incentive to good peers.

5.5. Overhead

The Overload of RC-ATP is evaluated by three metrics:

- The Number of Flooding Messages Per Query (NFMPQ): the average number of the replicas of each original query incurred by other peers' flooding-based propagating. It is a important character reflecting the scalability or efficiency of P2P networks. The scalability or efficiency decrease as the NFMPQ increasing.
- The network traffic: includes search query messages, response messages, file download messages and connection request messages.
- The Overhead of the Topology Adaptation (OTA): the ratio of connection request messages to the network traffic.

In Fig. 13 we compare the number of flooding messages per query of RC-ATP and APTP in condition of *ffraction* = 0.25 and of *ffraction* = 0.75. There is a peak of the NFMPQ of RC-ATP and APTP because when malicious peers are not recognized they keep connections with others, are always up and forward queries. The NFMPQ of

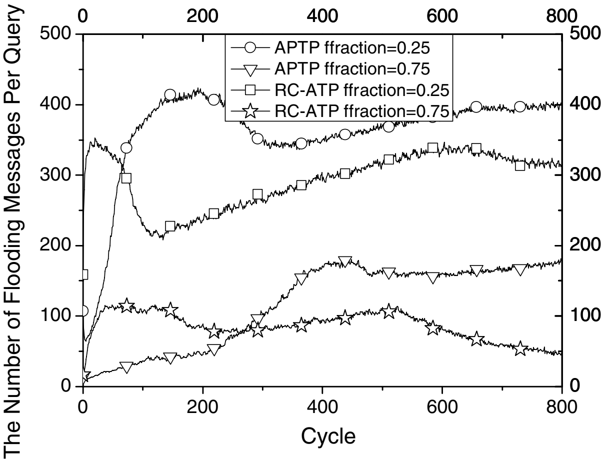


Fig. 13. The number of flooding messages per query.

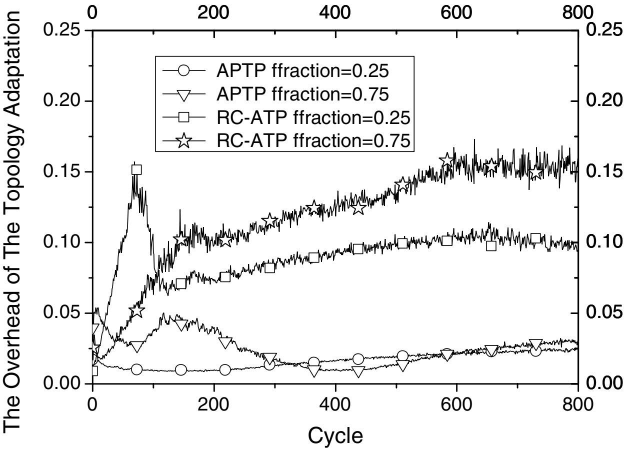


Fig. 15. The overhead of the topology adaptation.

RC-ATP is lower than that of APTP. This is attributed to peers dropping connections to non-reciprocal peers. The lower the NFMPQ is, the lower the cost of processing queries is and the larger the network productivity is. The NFMPQ in condition of $ffraction = 0.75$ is lower than that of $ffraction = 0.25$ because freeriders don't forward queries for others.

In Fig. 14, we show the network traffic of RC-ATP and APTP. The changing tendency of the traffic is similar with the NFMPQ. So the cost of the network working of RC-ATP is less than that of APTP. The most importance is RC-ATP is more efficient than APTP. As a result, the productivity of the RC-ATP is larger than that of APTP.

Fig. 15 plots the OTA of RC-ATP and APTP. In RC-ATP, there is a peak of the OTA in the initial stage in condition of $ffraction = 0.25$ while this is not the truth in condition of $ffraction = 0.75$. This is because when $ffraction = 0.25$, peers can quickly know the peers which may benefit it in future and send connection requests to those peers. But when $ffraction = 0.75$, peers can't locate the requested files efficiently and they only try to connect new neighbors when $|Fellow(i)| < \tau^i_{\min}$ or can't get responses for several observing windows. So the OTA in condition of $ffraction = 0.75$ is steadily increasing. The careful reader would notice the OTA has a tendency to decrease after cycle 654 in condition of $ffraction = 0.25$. It is the case that freeriders have little chance to get connected with other peers and begin to give up sending connection requests because they have been known by almost all the peers in the network. The larger *ffraction* is, the larger the overhead of the topology adaptation is.

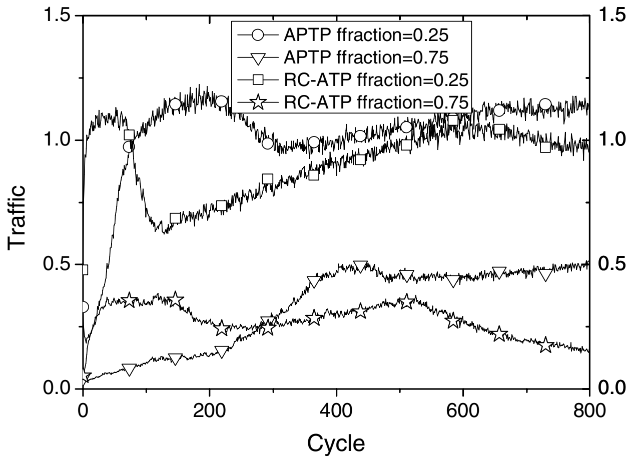


Fig. 14. The Traffic.

So the conclusion can be drawn that although the OAT of RC-ATP is a bit larger than that of APTP, it is a more promising mechanism than APTP because it works more efficiently with less cost.

6. Conclusion

In this paper, we discuss the necessity of taking into account the issues of freeriders or malicious peers to build robust and efficient P2P networks and propose a reciprocal capacity based adaptive topology protocol to maximize the network productivity. This protocol is based on the rational belief that a peer is only willing to maintain connections with those which will benefit it in future. Reciprocal capacity is defined based on the capacity of providing services and the capacity of recommending service providers, which are calculated according to the peers' behavior history. In addition, a response selection mechanism is proposed to reduce the probability of trying to download files from malicious peers. The resulting topology is a small world network where peers can communicate quickly. In RC-ATP good peers can locate authentic files within fewer hops and download authentic files with the larger probability of the success download compared with APTP. So it is more efficient than APTP. Due to the adequate connections between reciprocal peers, the network connectivity and the reachability between good peers are better in RC-ATP than in APTP under DAttack and BAttack of nodes. So RC-ATP is more resilient than APTP. Furthermore, it has the intrinsic incentive to good peers as they can efficiently get authentic files with the larger ratio of the success query and fewer hops compared with freeriders while malicious peers are eliminated from the network. In addition, good peers have a stable connection relationship with their neighbors while freeriders have to frequently change neighbors to get profit. Although the overhead of the topology adaptation is a bit higher, RC-ATP works more efficiently with less cost compared with APTP.

In RC-ATP, the reciprocal capacity is calculated by local interaction history. With the network scale increasing, the chance that peers are cheated by malicious peers is increasing. So we plan to study how to share opinions between peers and enable malicious peers to be known quickly. This would make RC-ATP more scalable. While shared opinion is scalable, it is vulnerable to collusion. If the reciprocal relationship between peers can be modeled as a graph, the maxflow algorithm can be used to deal with collusion. However, the maxflow algorithm needs long running time and is unfeasible in real networks. Currently, handling the collusion with acceptable cost in opinion sharing systems is still an open issue. Moreover, some peers may acquire high reciprocal capacity score by proving authentic files for a while, and then traitorously provide inauthentic files. How to adapt the observing window to avoid be attacked by such peers would be studied further.

References

[1] T. Condie, S.D. Kamvar, H. Garcia-Molina, Adaptive peer-to-peer topologies, in: P2P'04, August 2004, pp. 53–62, 25–27.

[2] I. Stoica, R. Morris, D. Karger, et al., Chord: a scalable peer-to-peer lookup service for internet applications, in: SIGCOMM 2001, San Diego, California, USA, 2001.

[3] S. Ratnasamy, P. Francis, M. Handley, et.al. A scalable content-addressable network, in: SIGCOMM 2001, San Diego, California, USA, 2001.

[4] B.F. Cooper, H. Garcia-Molina, Ad Hoc, Self-supervising Peer-to-Peer Search Networks: Technical Report, Stanford University, 2003.

[5] Q. Lv, S. Ratsnasamy, S. Shenker. Can heterogeneity make gnutella scalable?", in: First International Workshop on P2P Systems, 2002.

[6] Y. Chawathe, S. Ratnasamy, L. Breslau, et al., Making gnutella-like P2P systems scalable, in: SIGCOMM 2003, Karlsruhe, Germany, August 2003.

[7] Y. Liu, Zh. Zhuang, et.al., AOTO: adaptive overlay topology optimization in unstructured P2P systems, in: Globecom'03.

[8] Y. Liu, X. Liu, L. Xiao, et.al., Location-aware topology matching in P2P systems, in: Infocom'04.

[9] L. Xiao, Y. Liu, L.M. Ni, Improving unstructured peer-to-peer systems by adaptive connection establishment, Comput. IEEE Trans. 54 (9) (2005) 1091–1103.

[10] K. Sripanidkulchai, B. Maggs, H. Zhang, Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems, IEEE Infocom, San Francisco, USA, 2003.

[11] H. Kobayashi, H. Takizawa, T. Inaba, et.al. A self-organizing overlay network to exploit the locality of interests for effective resource discovery in P2P systems, in: Proceedings of the 2005 Symposium on Applications and the Internet (SAINT'05).

[12] E.A. Bernardo, A. Huberman, Free Riding on Gnutella Tech Rept: SSL-00-63, Xerox PARC, 2000.8.

[13] R. Bayya, et al. Economic models for management of resources in P2P environments, in: SPIE International Conference on Commercial Applications for High-Performance Computing,Computational Economics Press, Denver, USA , 2001.8, pp. 1–12.

[14] D. Wen, The Research on Trust-Aware P2P Topologies and Constructing Technologies, Ph.D. Thesis, National University of Defense Technology, PR China, October 2003.

[15] D. Heckerman, A Tutorial on Learning with Bayesian Networks Technical Report. 1995. <ftp://ftp.research.microsoft.com/pub/tr/tr-95-06.pdf.>.

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

[17] <http://p2p.stanford.edu/www/demos.htm.>.

[18] M. Schlosser, T. Condie, S. Kamvar, Simulating a file-sharing P2P network, in: FirstWorkshop on Semantics in P2P and Grid Computing, December, 2002.

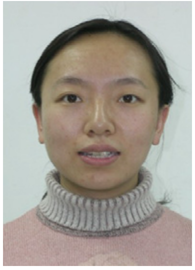
[19] S. Saroiu, P.K. Gummadi, S.D. Gribble, A measurement study of peer-to-peer file sharing systems, in: MMCN' 02, 2002.

[20] J. Scott, Social Network Analysis: A Handbook" London, Sage Publications, London, 1991.

[21] L. Freeman, Centrality in social networks: conceptual clarification, Social Networks 1 (1978/79) 215–239.

[22] R. Albert, A.L. Barabasi, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (2002) 47–97.

[23] R. Albert, H. Jeong, A.L. Barabasi, Error and attack tolerance of complex networks, Nature 406 (2000) 378–382.



Huirong Tian is a PhD candidate of the State Key Lab of Networking and Switching Technology at the Beijing University of Posts and Telecommunications. Her research interests are service management, peer-to-peer networks, and cooperative systems.

Shihong Zou received his Bachelor of Engineering degree in Computer Engineering from Nanjing University of Posts and Telecommunications (Nanjing, China) in 1999, and his Ph.D. degree in communication and information systems from Beijng University of Posts and Telecommunications (BUPT) in 2004. He is currently a lecturer in BUPT. His research interests include IP QoS, WLAN, mobile ad hoc networks and wireless sensor networks.

Wendong Wang is the professor of State Key Lab of Networking and Switching Technology at Beijing University of Posts and Telecommunications. From 1993 to 1996, he was invited many times to Alcatel Bell, Belgium for the research of the cooperative project and the technology intercommunion. He has been in Canada as a visiting scholar from 2000 to 2001. He is the member of expert group in national 863 programs on communications. He has published more than 80 papers in the field of communications. His research interests are the QoS control and the management of the next generation internet service.

Shiduan Cheng is the professor of State Key Lab of Networking and Switching Technology at Beijing University of Posts and Telecommunications. From 1984 to 1987 and in 1994 she twice joined Alcatel Bell, Belgium as a visiting scholar. From 1992 to 1999 she was the head of The Switching and Networking Expert Group in 863 programs, a national high-tech R&D plan organized by The Ministry of Science and Technology of China. She has published more than 100 papers and several books in the field of telecommunications. Her research interests cover ISDN, ATM, TCP/IP, switching software, protocol engineering, traffic engineering, network performance, QoS, security and survivability. Currently she is working on the QoS control, the measurement and management of the next generation internet service, ad hoc and mobile networks.