CCLBR: Congestion Control Based Load Balanced Routing in Unstructured P2P Systems

Xiang-Jun Shen, Qing Chang, Lu Liu, *Member, IEEE,* John Panneerselvam, and Zheng-Jun Zha, *Member, IEEE*

***Abstract*—Given the growing popularity of the peer-to-peer (P2P) network systems in the recent years, efficient query routing under highly dynamic environments is still lacking in several P2P network systems. In response to this challenge, this paper proposes a new churn-resilient system to find alternative routing paths for the purpose of balancing the query loads under higher network churns and heavy workloads, ultimately to improve the search efficiency. Two novel methods are devised to balance the network query loads among both inter- and intra-group level peers. Firstly, a resource grouping and a rewiring method is proposed to spontaneously organize and cluster the peers having same resources together. This strategy facilities the peers to evolve the network into a cluster-like topology and balances the query loads among the inter-group peers. Secondly, a collaborative Q-Learning method is proposed to balance the query loads among the intra-group peers in order to intelligently avoid queries being forwarded to the congested peers in the network. Experiments conducted under dynamic network scenarios demonstrate that our proposed method achieves better search performances with a more balanced network load than the existing methods, and further exhibits higher robustness and adaptability under higher network churns and heavy network loads.**

***Index Terms*—Congestion control, load balancing, query routing, collaborative Q-learning, unstructured P2P systems**

I. Introduction

P2P (Peer-to-Peer) networking systems are growing in popularity and are being deployed in a wide range of Internet applications [1], [2] such as content delivery, file sharing and multimedia streaming etc. Based on their resource utilization techniques and peer linking strategies, P2P systems are classified into two broad categories such as structured networks and unstructured networks. Query load balancing is one of the most prevalent issues in both the two types. Unstructured P2P networks are more vulnerable to the query load effects since the peers in the network often lack a complete knowledge of the network structure. This causes peers to forward queries to distant and adjacent peers, and such forwarding strategies often lead to load balancing issues [3] in unstructured P2P networks.

Xiang-Jun Shen, Qing Chang are with School of Computer Science and Communication Engineering, JiangSu University, Zhenjiang, China (e-mail: xjshen@ujs.edu.cn, 939302304@qq.com).

Lu Liu is with the Department of Computing and Mathematics, University of Derby, Derby, UK and School of Computer Science and Telecommunications Engineering, Jiangsu University, Zhenjiang, China (e-mail: l.liu@derby.ac.uk)

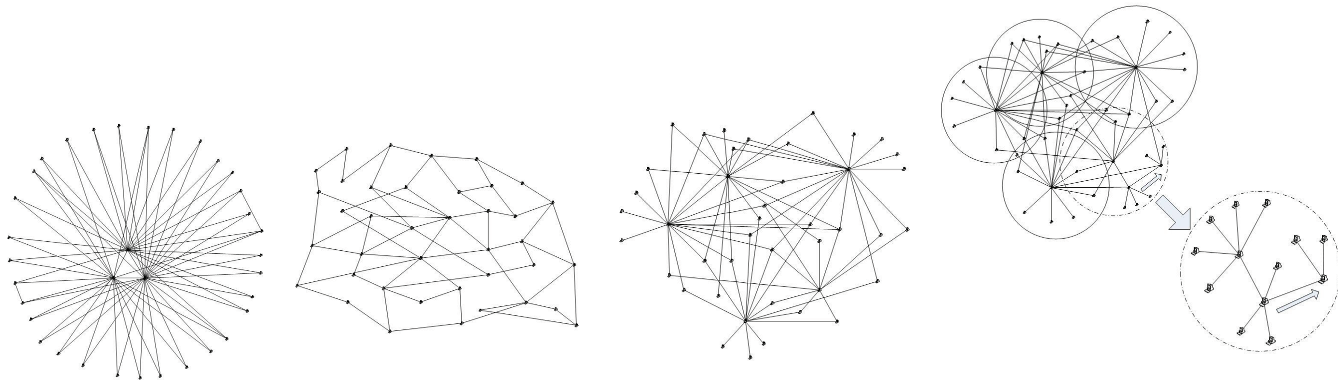
John Panneerselvam is with School of Computing and Mathematics, University of Derby, Derbyshire, UK(e-mail: j.panneerselvam@derby.ac.uk)

Zheng-Jun Zha is with School of Information Science and Technology, University of Science and Technology of China, Hefei, China (e-mail: junzzustc@gmail.com)

Recently, three different types of overlay topologies (as shown in Fig.1) have been proposed [4], [5], [6] to be deployed under various levels of query burdens in Unstructured P2P networks. The star-like overlay topology functions as a Client/Server (CS) model, which is more suitable for light-level query workloads. But, the star topology loses efficiency with increasing number of network queries, as the central peers are usually overloaded by the immense query loads. Thus, heavy query loads rapidly deteriorates the search efficiencies of the star topology. The randomized overlay topology performs better than the star topology under heavy workloads (increasing queries) by effectively sharing the search responsibilities among all the peers in the network. Though, locating the required resources is still challenging in the randomized topology, which insists the need for further improvements in their search strategies. The cluster-like overlay topology performs well under medium-level query workloads, but the trade-off between search efficiency and load balancing among the peers is not often satisfied. To this end, a number of algorithms [4], [5], [6], [7], [8], [9], [10], [11] have been proposed recently for the purpose of balancing the trade-off between search efficiency and load balancing. A common approach of most of these algorithms is to evolve the network into a cluster-like topology from the randomized topology. But, most of the existing approaches are focused on balancing the query loads only among the inter-group level peers. In general, ignoring the query loads among the intra-group level peers will significantly affect the overall routing efficiencies of the peers in the unstructured P2P networks.

With this in mind, this paper proposes a novel approach of clustering the network peers both at the inter- and intra-group levels based on the resources contained within the peers. The characteristic features of our proposed system are described as follows:

1. Achieving the network topology evolution through a resource grouping and a rewiring strategy in order to ensure optimum clustering of the network peers. Resource grouping strategy clusters the peers having the same resources together. This allows the network peers to establish reconnection with high capacity peers, so that the evolved network will resemble a local cluster-like topology in every formed cluster. By this evolution technique, we balance the query loads among the inter-group level peers and improve the efficiency of the resource locating phase. In the meantime, the rewiring strategy prevents overloading of the local clustered peers by the way of disconnecting their links from the over-



(a) (b) (c) (d)

Fig. 1. Network topologies under different query workloads. (a)Star-like topology (light), (b)Randomized topology (heavy), (c)Cluster-like topology (medium), (d) Our proposed topology.

loaded peers accordingly, as illustrated in Fig.1(d) (where peers within circles represent the same group).

2. Controlled routing whilst balancing the query load among the intra-group peers is achieved by employing a collaborative Q-Learning [12] technique among the peers. This proposed collaborative Q-Learning effectively learns the network parameters such as processing capacity, the number of peer connections, and the number of resources contained within the peers, along with their state of congestion in the network. By this technique, peers are avoided to forward queries to those congested peers in the network. Thus, queries are always forwarded to the balanced intra-group peers, as illustrated in Fig.1(d) (where the arrows depict the forwarding direction of the queries among the intra-group peers in the right side circle).

Our simulation results demonstrate that the requested resources are located more quickly whilst achieving query load balancing in the entire network. Also, our method exhibits more robustness and adaptability under both higher network churns and heavy query workloads in comparison to the random walk method and the DANTE system [6]. The simulation results also illustrate that the proposed system achieves better search performances than the DANTE system which is also a self-organized evolving network. Through our simulation results, we prove the significance of our proposed dynamic overlay topology method in terms of scalability, robustness and churn resilience.

The rest of the paper is organized as follows: Section II presents the related recent research works of load balancing algorithms in P2P networks. Section III formulates our proposed method of load balanced routing based on congestion control in P2P networks. Performance evaluations and our simulation results are presented in section IV. Finally, we draw conclusions in section V.

II. Related Work

In general, the load balancing mechanisms perform distinctively in structured and unstructured P2P networks. Structured P2P networks have a dedicated overlay network design and uses DHTs to establish links between stored resources and address of the peers. DHT search algorithms [1] in structured networks such as Chord, CAN and Pastry perform efficient resource discovery based on the knowledge of the hash keys. But a notable shortcoming of such algorithms is their huge communication costs incurred under higher network churn, since the ID space of their DHTs is partitioned among the peers and their resources. This causes structured P2P networks to become smaller and inflexible.

In unstructured P2P networks, no such extra structure for the space partition is required. This makes unstructured P2P networks to be more flexible and extensible. But the shortcomings of the unstructured P2P networks are evident in their complex resource locating strategies, as queries can only be forwarded to geographically adjacent peers. Also, the network topologies of some of the advance level resource searching methods such as flooding [13] and random walk [14] often lead to load balancing issues among the peers. Possible ways to overcome such issues are to improve the efficiencies of a notable search algorithm called random walk sampling [15], and further by guiding the search methods through information collections [16], [17].

Furthermore, load balancing methods can also be effectively utilized to solve the issues of imbalanced workloads in unstructured P2P networks. Some of the research works used super peers [18], [19] to control the broadcasted queries over the network. But this approach might suffer from single-point-of-failure issues and the scalability of the network is limited with the departure of these super peers [20]. Identifying the free ridings of the uneven popular files over the peers could transform a random network into a star topology [21], [22] in flat networks, which would help to dynamically adapt the desired network overlay topologies. Such strategies of dynamically adapting the network topologies are found to be deployed in some of the recent works. For instance, Merino et al. [6] introduced the DANTE network, which employs a self-organized reconnection mechanism to form a balanced cluster overlay topology. Pournaras et al. [8] proposed the

TABLE I
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
P_i	the initiator peer that initially sends a resource grouping message
CP_i	the set of candidate peers chosen for establishing reconnection with P_i
PRA	Peer Resource Attractiveness, which is defined for the purpose of selecting the appropriate peers in CP_i
C_i	the maximum number of queries that P_i can process per micro-second
$\chi(i, k_c)$	the degree of peer connectedness of P_i
CL	Congestion Level, which is used to evaluate the congestion state of a peer
$Q_i(t)$	the number of queries waiting in the input queue of P_i at time t
U	the threshold defined to predict the congestion caused by overloading the peers with more number of queries
m_t	the threshold used to control the number of reserved queries
$Q(s, a)$	an action-value function, which is used to evaluate the effectiveness of the chosen action a under state s
$R(s)$	the reward function of state s
AP_s	the positive attractiveness of P_s

ERGO network with a rewiring strategy by utilizing virtual servers to monitor the workload of the peers. When overloaded peers are identified in the network, ERGO rewires some of the incoming links of the overloaded peers to the under-loaded peers. AVMON [9] is another network that monitors the content discovery and collusion of the network peers. Mashayekhi and Habibi [10] proposed a framework that maintains limited size routing indexes for combining search and trust data, in order to forward queries to the most reputable peers. LARD (Learning Automata-based Resource Discovery) [11] is another unstructured P2P network in which each peer chooses a communication link to route the query towards resource providers. If the selected route is shorter than the average length of the previously selected routes, this algorithm rewards the selected route, otherwise the chosen route is penalized. Thus, this algorithm converges to the route having the minimum expected length. The strategies of using small-world overlay topologies are introduced in the works of [7], [23]. Merugu et al. [7] used two types of links such as the short links for connecting closer peers and the long links for connecting randomly chosen peers. Wu et al. [23] proposed a similar mechanism to maintain a state-based shortcut list in the peers. Efficient search is achieved in this method by using either the links or the state information.

Based on above discussion, existing cluster-like topology mechanisms are found to be focused at clustering only the inter-group level peers based on the resources contained within such peers. To this end, we propose a clustering approach of clustering both inter- and intra-group level peers based on their corresponding resources. The strategy of clustering peers having same resources together at the inter-group levels facilitates effective resource discovery and load balancing among such peers. Furthermore, our approach also clusters the peers at the intra-group level, which further increases the effectiveness of the cluster-like topology structure. Among the intra-group level peers, queries are always forwarded to the balanced peers through a collaborative Q-Learning method, which is a popular reinforcement learning (RL) method [24] being employed to learn the network environments. In our proposed approach, this RL technique is used to monitor the state of the peers in the network, whilst achieving load balancing among the intra-group level query loads.

III. CONGESTION CONTROL BASED LOAD BALANCED ROUTING

In this section, we introduce our proposed churn-resilient system for congestion control based on load balanced routing in detail. In our proposed system, query routings are balanced among both inter- and intra- group level peers. The first subsection details our approach of balancing the query loads among the inter-group level peers. While in the next subsection, our novel query routing control strategy based on collaborative Q-Learning among the intra-group level peers is introduced. To ease the descriptions of the following subsections, we first list several notations and their descriptions in Table I.

A. *Resource grouping and rewiring*

Resource grouping and rewiring strategies in our proposed approach helps quickly locating the required resources in the network. In the resource groping process, it is assumed that peers with same resources may often have similar interests. In some of the P2P applications such as file sharing and video on-demand, peers those involved in the downloading process of the same file or in the buffering process of the same video might exhibit very similar interests on other resources in the network. Peers maintaining close contacts with other peers that have similar interests, will search resources quicker. Similar strategy can be found in the BitTorrent protocol [25], which applies the swarm technique for searching the target resources. In our proposed network, no server is used and the resource grouping is triggered periodically by the individual peers in order to enable themselves to form a cluster-like topology.

Let the initiator peer be P_i, which initiates a new resource grouping process for searching resource j. Since P_i has n different resources, it may belong to n different resource groups. Now, P_i triggers a new grouping process by launching a look-for-peer message for the purpose of collecting the list of peers having same resources to that of P_i. This look-for-peer message of P_i is a quad-tuple composed of $\{P_i, CP_i, j, TTL\}$, where P_i is the initiator peer sending the message, CP_i is the set of candidate peers chosen for establishing reconnection with P_i for searching resource j, and TTL is the Time-to-Live parameter which is used to control the number of hops that peers can traverse through the network. Initially the candidate peer set P_i is empty when the look-for-peer message is broadcasted over the network through the random walk method [14] in order to collect the list of candidate peers. When a peer P_k receives a look-for-peer message, the value of the TTL is decreased by 1 in the received message, and P_k checks its resources for matching the resource request. If P_k has the requested resource j, it appends itself to the set of candidate peers CP_i. Otherwise, P_k will not be added to the set of candidate peers CP_i. P_k then employs the random walk method to select one of its neighbors in order to forward the look-for-peer message. This neighbor then repeats the above process and allows the look-for-peer message to be broadcasted through the network till the value of the TTL decreases to zero. Finally, the look-for-peer message is sent back to the initiator peer P_i.

When P_i receives the set of candidate peers CP_i whilst searching for resource j, it chooses some of the peers in CP_i in order to establish links with them. Now, we introduce a parameter named Peer Resource Attractiveness (PRA) for the purpose of selecting the appropriate peers in CP_i . This PRA is a composite composed of peer degree, processing capacity and the number of resources contained within a peer. PRA is defined as follows:

$$PRA_i = \chi(i, k_c) \times C_i \times n_i$$
$$\chi(i, k_c) = \sum_{h=1}^{k_c} \frac{N(i,h)}{h^\sigma} \tag{1}$$

where, $\chi(i, k_c)$ denotes the degree of the peer connectedness of P_i, $N(i, h)$ denotes that the counting degrees of the neighboring peers are h hops away from P_i, and k_c is the radius parameter for counting P_i's neighbor peers. The parameter σ (used is to control the value of h^σ) is a weight used to control the peers at different hop distances away from P_i. The higher the value of σ less is the impact of P_i on the remote peers. The higher value of $\chi(i, k_c)$ implies a higher peer connectedness of P_i. C_i is the maximum number of queries that P_i can process per micro-second. N_i is the number of resources contained within p_i. A_i denotes the effects of the peer connectedness, processing capacity and the number of resources contained within a peer. The larger value of A_i reflects the higher processing capacity of the peers along with its higher peer degrees. The higher the value of A_i higher is the number of resources contained within the corresponding peers. Obviously, the larger value of A_i insists the increased attractiveness of P_i to be a reconnecting peer.

Guided by this PRA parameter, the reconnection process of P_i with the selected peers in the set CP_i is explained as follows:

a. P_i chooses a reconnecting peer P_a (having the highest value of PRA) from the set CP_i, as defined in formula 1. Then P_a is removed from the set CP_i;
b. P_i chooses one of its neighbor peers for the purpose of disconnecting it. The peer link of this chosen neighbor peer should be greater than 2, in order to prevent the peer from becoming completely isolated. A neighbor peer is chosen for disconnection only if the corresponding peer does not own the resource j. If such a neighbor peer does not exist, P_i then chooses its weakest neighbor for disconnection. Now, the chosen peer is marked as P_d;
c. P_i then sends a connection request to P_a. Once P_a accepts this connection request, P_i disconnects P_d and connects P_a.

The above process is repeated until either the candidate set CP_i becomes empty or the PRA value of each peer in CP_i becomes weaker than the minimum PRA value of the neighbor peers of P_i.

From formula 1, a peer with larger value of PRA can process more forwarding queries and it is more likely to be the local central peer owning more resources with higher peer degrees. This makes the search process easier among the inter-group level peers. Meanwhile, more number of peer-links and increasing number of forwarding queries causes peers to be vulnerable for network congestion. The rewiring strategy is now applied to disconnect some of the neighbors of the congested peers in the network. This strategy decreases the number of query routings and helps the congested peers to become normal in the network. Now, a parameter named Congestion Level (CL) is introduced to evaluate the congestion state of a peer. The value of CL of a peer P_i at time t is computed as:

$$CL_i(t) = \frac{1 + Q_i(t)}{C_i} \tag{2}$$

where, $Q_i(t)$ denotes the number of queries waiting in the input queue of P_i at time t. When a peer is processing a query, the forthcoming forwarding queries are usually put into the waiting queue of that corresponding peer. Formula 2 implies the waiting time that a query would spend if it is forwarded to P_i. The higher the value of $CL_i(t)$ higher is the congestion state of a peer. We use a threshold U to assert a peer to be congested.

When $CL_i(t)$ is larger than U, the corresponding peer is considered to be over-loaded. A time cycle is used to trigger the rewiring strategy among the peers in the network. When a peer reaches the overloaded state, it triggers rewiring. By this rewiring technique, a given peer disconnects some of its neighbors randomly and connects those disconnected peers to other such peers having same resources. The number of neighbor peers to be disconnected is computed using the following rules.

When the peer P_i is overloaded, some of its neighbor links are disconnected and so the number of queries forwarded to P_i will decrease. Therefore, we believe that the number of queries being forwarded to a peer is proportional to the number of neighbor connections of that corresponding peer. Since U is the threshold defined to predict the congestion caused by overloading the peers with more number of queries, and C_i is the maximum number of queries that P_i can process per micro-second, the maximum number of queries that can be processed by a peer without being regarded as overloaded in the waiting queue is given by $U \times C_i - 1$. After a peer disconnecting its neighbors, $m_t \times U \times C_i - 1$ queries are assumed to be reserved for that peer. m_t is the threshold used to control the number of reserved queries. In our experiments, we set the value of m_t as 0.8, which means that 80% queries should be reserved for a peer. The number of links to be disconnected D_i is deduced using the following equation.

$$\begin{aligned} \frac{Q_i(t)}{m_t \times U \times C_i - 1} &= \frac{N_i}{N_i - D_i} \\ D_i = \lceil \frac{N_i \times (Q_i(t) - (m_t \times U \times C_i - 1))}{Q_i(t)} \rceil \end{aligned} \tag{3}$$

where, N_i is the current number of connected neighbor links of P_i. $\lceil . \rceil$ is a ceiling function.

In this way, the overlay topology is updated in order to enable peers to establish close connections with other such peers having same resources. This resource grouping and rewiring strategy balances the query loads among the inter-group level peers and helps the evolution of the entire network into a cluster-like topology from a randomized overlay topology. Meanwhile, this system also drives the peers of higher processing capacity to gain higher connection degrees than the weaker peers within the same group. The peers with higher connection degrees thus have higher congestion probabilities. Therefore, it is necessary to maintain an optimum level of query loads in every peer which will be dealt in the following section. Our collaborative Q-Learning method is introduced in the next subsection to achieve the intra-group load balancing.

B. Congestion control routing through collaborative Q-learning

In order to identify the congested peers and to avoid queries being forwarded to such congested peers, Q-Learning method (which is a method of Reinforcement Learning) is applied to monitor the state of the peers in the network. In this approach of Reinforcement Learning (RL) [24], RL agents learn by interacting with their environment and by observing the outcomes of such interactions. Q-Learning [12] is an Off-Policy algorithm of RL used for temporal difference learning. It uses an action-value function Q to directly approximate the optimal action-value for an arbitrary target policy. The one-step Q-Learning model is defined as follows:

$$\begin{aligned} Q^{local}(s,a) &= R(s) + \gamma max_{a'} Q(s', a') \\ Q^{new}(s,a) &= Q(s,a) + \alpha Q^{local}(s,a) \end{aligned} \tag{4}$$

where, $Q(s,a)$ is an action-value function. $R(s)$ is the reward. α is the learning rate which is set between 0 and 1. γ is the discount factor, also set between 0 and 1. This parameter of γ considers that future rewards are worth less than the immediate rewards. s is the current state and a is the action of the current peer in s. s' is the next state, and a' is the reaction of the peer in s'. In our network, state s contains the current peer which transmits the routing messages to its neighbors. State s' will encompass one of the neighbors of the peer in s. This neighbor will receive the messages from the peer in s. So, a is the action of the current peer, which is selecting one of the neighbors of the peer in s in order to transmit the routing messages. Meanwhile, a' is the action of the peer in s', which is selecting the neighbor peer of s' to transmit the routing messages. $max_{a'}$ is the maximum reward that can be achieved in the next state s.

To evaluate the congestion state of the peers in the network, state information relevant for the routing process such as processing capacity, number of connections and number of resources are monitored. The parameters encoded in the $R(s)$ function reflect the basic state of the peers in the network. $R(s)$ function is defined as follows:

$$\begin{aligned} R(s) &= \sum_{i=0}^{\infty} \gamma^i \frac{AP_s}{N_s} \\ AP_s &= C_s \times \chi(s, k_c) \end{aligned} \tag{5}$$

where, C_s and $\chi(s,k_c)$ are defined in formula 1. $\chi(s,k_c)$ denotes the degree of the peer connectedness of P_s and C_s is the maximum number of queries that P_s can process per microsecond. By this way, AP_s denotes the positive attractiveness of P_s. The large value of AP_s implies the higher processing capacity of P_s along with its higher peer degrees. Finally, N_s is the number of resources contained in the peer P_s. N_s is a negative factor in the formula. More the number of resources contained in P_s more is the number of forwarding queries required to pass through. This implies that the bandwidth allocation for a single resource is considerably low.

From the above formula, it can be understood that more number of rewards will provide larger values of AP_s to the peers. Also peers having more number of neighbors and connections can process more forwarding queries than others. On the other hand, such peers with increased number of connections can easily cause congestion in the network. In order to balance this effect, we regularize the basic Q-Learning model by adding the parameter of Congestion Level (CL), which is defined in formula 2. The modified Q-Learning model is described as follows:

$$\begin{aligned} Q^{new}(s,a) = Q(s,a) + \alpha Q^{local}(s,a) + \\ \beta \times I(U - CL_{s'}(t)) \times CL_{s'}(t) \end{aligned} \tag{6}$$

where, $I(x) = \begin{cases} +1, & x > 0 \\ -1, & x \leq 0 \end{cases}$ is an indicator function. This function gives a positive sign when a peer is in normal state and a negative sign when the peer is overloaded. Thus our proposed model incorporates the effects of congestion state of the peers whilst forwarding query loads to the peers in the network.

From formula 6, it can be observed that the computation of the Q-values of the peers considers the processing capacity, number of connections and number of resources, along with the congestion state of the peers. In this way, query routings are controlled by the collaborative Q-Learning among the intra-group level peers.

IV. Performance Evaluation

A. Simulation setup

Our simulation environment is composed of $10,000$ peers, developed based on Gnutella in Python 2.6. Initially, every peer in the network is assigned with 10 neighbors on average. Assuming that there are n queries in the peer P_i, the amount of time spent in processing n queries is $\frac{n}{C_i}$, where C_i is the

TABLE II
DISTRIBUTION OF PEER PROCESSING CAPACITY IN THE SIMULATED NETWORK.

Percentage of peers (%)	Processing capacity C_i
20	0.1
45	1
30	10
4.9	100
0.1	1000

number of maximum queries that can be processed by the peerP_i per micro-second, as mentioned in formula 1. The heterogeneity characteristics of the P2P networks in terms of the processing capacity are presented in table II. This distribution is obtained from the measure of Gnutella reported in [26]. Since the network load is actually caused by the resource queries launched by the peers, we assume that for all the peers in the network, the time interval between two successive queries is equal and is termed as the time between searches (tbs). For instance, $tbs = 5s$ reflects that $2,000$ queries are generated per second in the system on average.

$1,000$ object resources are used in the network, with each object having duplicates. The number of object duplicates is determined by the popularity of the objects using a Zipf-like distribution property [6]. We randomly distribute these objects to different peers in the network. The replication factor of the most popular object is 50%, implying that the object has $5,000$ duplicates in our system. On the other hand, the least popular object has only a replication factor of 0.5%. Now, 5 random walkers are dispatched for each query in the network to locate the requested resources. The TTL of the query request is set to 8 and the TTL of the look-for-peer message is set to 30. This value is high enough to obtain a good sampling of the network.

The parameter settings of our proposed formula are described as follows: The two parameters k_c and σ of formula 1 are set to 2 and 1, respectively. It is evident that the network bandwidth consumption is attaining growth with an increasing k_c. The parameter m_t used in formula 3 is set to 0.8, which means that 20% connections are supposed to be disconnected from an overloaded peer. The parameter U is set to 1.1, implying that the peers are allowed to have 10% more queries than their actual capacity of processing maximum number of queries per micro-second. Regarding formula 4-6, γ is set to 0.3. α is set to 0.3 and β is set to 0.5 respectively.

Fig.2 shows the computing costs of random walk, DANTE system and our proposed system, respectively. The real CPU time is recorded in time steps of 3 hours in a stable network to reflect the computing costs of the three methods. It is evident from Fig.2 that the random walk method achieves the best computing costs of 48 hours, since it do not incur any extra costs for transmitting routing messages. The DANTE system achieves the next better computing costs of 61 hours, since it uses a time cycle to adjust the network periodically. Our proposed system achieves worse computing costs of 68 hours (11.5% extra CPU time than the DANTE system). This extra CPU costs of our proposed system is because our system updates the Q values of the peers.

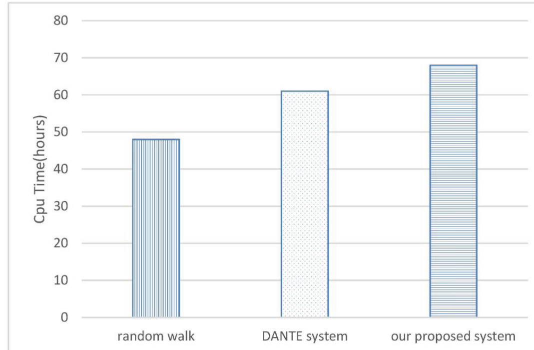


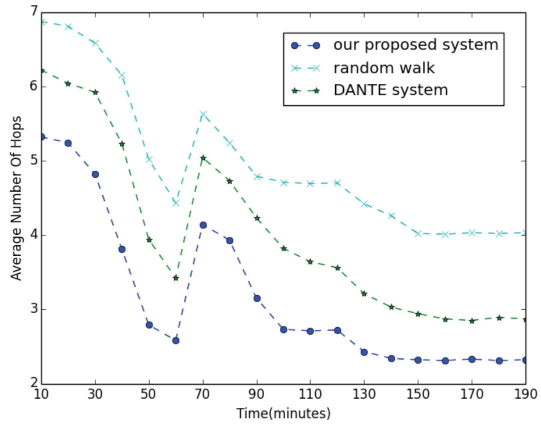
Fig. 2. CPU time consumption.

In order to evaluate the performance of our proposed system, we simulate the network under various network settings. Further, we evaluate the efficiency of our proposed system against the random walk method, which is a standard search method in P2P networks, and the DANTE system [6], which also dynamically adapts to the network overlay topology for improving the search performances in P2P networks. In subsection IV-B and IV-C, the network is simulated under higher churns and heavy workloads in order to test the robustness and adaptability of our proposed system. In subsection IV-D, we design a more realistic simulation environment with a combination of higher churns and heavy workloads. We evaluate our proposed system against the DANTE system to test the efficiencies of our proposed method in such a realistic scenario.

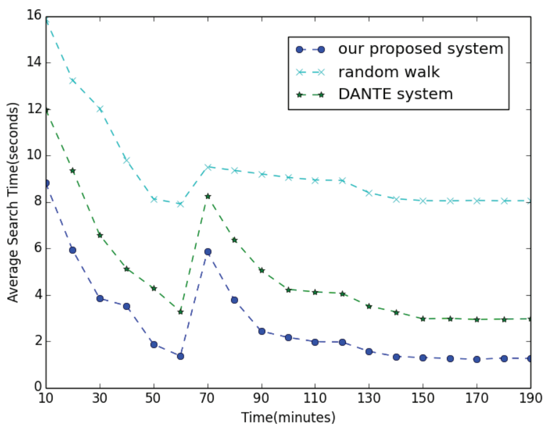
B. Network performance evaluation under higher churns

In this subsection, network performances under higher churns are evaluated. In order to observe the network performances under higher churns, peers with a processing capacity of $1,000$ are set to leave the network since they have the best processing capacity with more links and forwarding queries. Thus, such peers are more likely to become congested under higher network churns. Therefore, in our simulation settings, all the peers of processing capacity $1,000$ leave the network at the 60^{th} minute simultaneously and their links are redirected randomly to establish connections with other peers in the network. All these peers re-enter the network again at the 120^{th} minute. Between 60^{th} minute and 120^{th} minute, network performances under higher churns are observed.

Firstly, the search performances based on the average number of hops and the average search time are evaluated. Fig.3 shows the timely behavior of the network search performances of our proposed system, the DANTE system and the random walk method from the 10^{th} minute to the 190^{th} minute respectively. Fig.3(a) depicts the average number of hops of our proposed system, the DANTE system and the random walk method respectively. During the first 60 minutes, the network is static, i.e. no peers enter or depart the network. It is observed from the Fig.3 that the average number of hops among all the



(a)



(b)

Fig. 3. Timely behavior of the network search performances of our proposed system, the DANTE system and the random walk method under higher churns. (a)Average number of hops, (b)Average search time.

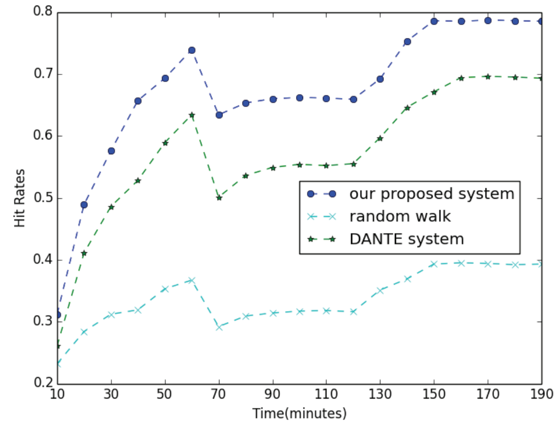


Fig. 4. Timely behavior of hit-rates of our proposed system, the DANTE system and the random walk method under higher churns.

three system is decreasing, and this decrease is sharper in our proposed system for the first 60 minutes. This demonstrates that our network system effectively evolves into a cluster-like topology and facilitates balanced routing by employing resource grouping and collaborative Q-Learning methods in such a stable network.

Between 60^{th} minute and 120^{th} minute, the network is subjected with higher churns and the peers of processing capacity $1,000$ leave the network simultaneously at the 60^{th} minute and they re-enter later at the 120^{th} minute. Between 60^{th} minute and 70^{th} minute, the average number of hops in all the three system is observed to be increasing, and this increase in our system is observed to be more than both the DANTE system and the random walk method. This is because the peers of higher processing capacity are subjected to be clustered in our system, and such clustered peers have greater impact on the forwarding queries. While, the random walk method shows a little increase in the average number of hops since the forwarding queries are sent randomly. This implies that our system dynamically adapts the network topology when the peers of higher processing capacity leave the network. While between 70^{th} minute and 120^{th} minute, the average number of hops of all the three methods is observed to be decreasing again. When the peers of processing capacity $1,000$ enter the network again at 120^{th} minute, we observe continues decrease in the average number of hops of both the other two methods. This is because when the network becomes stable again, and our proposed system effectively learns this change in the network topology and decreases the average number of hops to a lower level of 2.3, than those of the DANTE system and the random walk method. Fig.3(a) demonstrates that our proposed system effectively adapts to higher network churns when the peers of higher processing capacity leave the network rapidly. The average number of hops of our proposed system is less than those of the DANTE system and the random walk method under higher network churns, during the time between 60^{th} minute and 120^{th} minute.

Fig.3(b) illustrates the average search time performances of our proposed system, the DANTE system and the random walk method, between when the peers of processing capacity $1,000$ leave the network simultaneously at 60^{th} minute and enter the network again at 120^{th} minute. It can be observed that under higher network churns, our proposed system exhibits better adaptability to the changes in the network topology. The average search time of our proposed system is achieved quicker than the DANTE system and the random walk method at all times.

Secondly, the success of hit rates in the resource locating process are evaluated under higher network churns. Fig.4 shows the timely behavior of the hit-rates of our proposed system, the DANTE system and the random walk method respectively. For the first 60 minutes, no peers depart the network and the hit rates are observed to be increasing in all the three methods, especially in our proposed system. Between 60^{th} minute and 120^{th} minute, when the peers of processing capacity $1,000$ leave the network simultaneously, the hit rates of both the other two methods are observed to be decreasing until the 70^{th} minute and increasing thereafter. During this time, the hit rates of our proposed system are

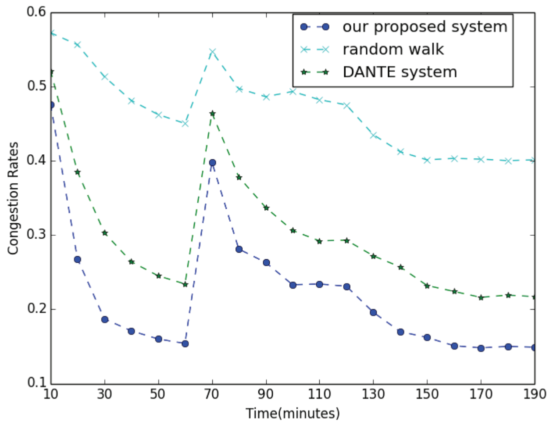
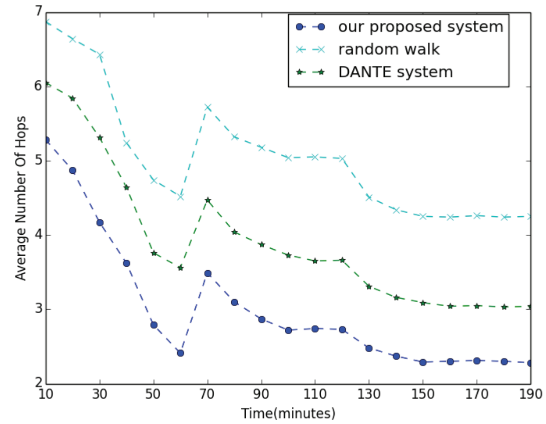
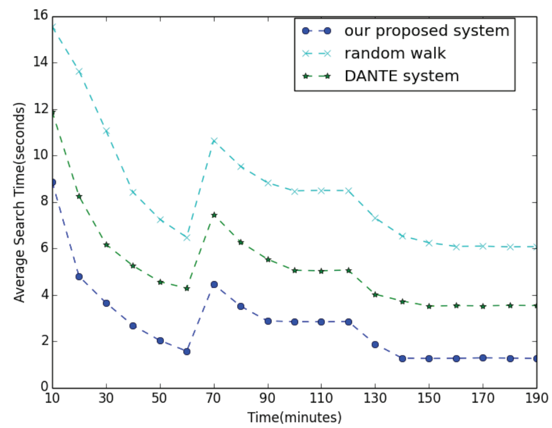


Fig. 5. Timely behavior of congestion rates of the peers of our proposed system, the DANTE system and the random walk method under higher churns.



(a)



(b)

Fig. 6. Timely behavior of the network search performances of our proposed system, the DANTE system and the random walk method under heavy workloads. (a)Average number of hops, (b)Average search time.

observed to be higher than those of the DANTE system and the random walk method. This demonstrates that our proposed system can rapidly adapt the changes in the network topology in order to form a clustered network again. Thereby, the success of hit rates in our proposed system is higher than those of the random walk method and the DANTE system. Meanwhile our proposed system has the ability to achieve balanced routing by employing the collaborative Q-Learning method among the intra-group level peers. Our system also outperforms the DANTE system which evolves into a clustered network without learning the network.

Finally, the congestion rate of the peers in the entire network is evaluated, as shown in Fig.5. When the peers of processing capacity $1,000$ leave the network simultaneously at 60^{th} minute, the peer congestion rates of all the three methods are observed to be increasing between 60^{th} minute and 70^{th} minute. This implies that the peers of higher processing capacity have greater impacts on the forwarding queries. Since our proposed system and the DANTE system evolves into clustered networks, the higher processing capacity peers have a greater impact on the forwarding queries in both the systems than in the random walk method. This can be asserted from the higher peaks evident between 60^{th} minute and 70^{th} minute in our proposed system and the DANTE system. After 70^{th} minute, the peer congestion rates of all the three methods are observed to be decreasing, especially in our proposed method. This illustrates that our proposed method effectively adapts to the changes in the network topology by employing the resource grouping and rewiring methods. Our routing control method by collaborative Q-Learning reduces the peer congestion rates more than those of the DANTE system and the random walk method. Thus in our system, routing messages are increasingly forwarded to the un-congested peers in the network.

C. Network performance evaluation under heavy workloads

In this subsection, network performances under heavy workloads are evaluated. In our simulation settings, query workloads in the network are increased at the 60^{th} minute and the workloads are set to normal again at the 120^{th} minute. Now, the network performances under heavy workloads are observed between 60^{th} minute and 120^{th} minute. In our simulations, the heavy workloads are generated in the network by setting the parameter $tbs = 0.5$, which means that the network generates 10 times more queries in the network than that of the normal workload condition where the tbs is 5.

Firstly, the search performances based on both the average number of hops and the average search time under heavy workloads are evaluated, as shown in Fig.6. During the first 60 minutes, both the average hops and the average search time are observed to be decreasing in all the three methods, as there are no heavy workloads in the network during this time. At the 60^{th} minute, our proposed system achieves lower average hops and average search time than both the DANTE system and the random walk method respectively. When queries are imposed ten times more in the network at the 60^{th} minute, both the average hops and the average search time are observed to be increasing in all the three methods until the 70^{th} minute and decreasing thereafter. During this time, both the average hops and the average search time of our proposed system are observed to be recovering more quickly and are lower than those of the DANTE system and the random walk method. Fig.6 illustrates that our proposed system can achieve better search performances under heavy workloads by employing routing control among both inter- and intra- group level peers.

Secondly, the success of the hit rates in the resource locating process is evaluated under heavy workloads. Fig.7 depicts the timely behavior of the hit-rates of our proposed system, the DANTE system and the random walk method under heavy workloads respectively. The hit rates are observed to be increasing in both the other two methods during the first 60 minutes, as there are no heavy workloads in the network. At the 60^{th} minute, our proposed system achieves higher hit rates than both the DANTE system and the random walk method, which demonstrates the efficiencies of our proposed system in a stable network. When queries are imposed ten times more in the network at the 60^{th} minute, the hit rates of both the other two methods are observed to be decreasing until the 70^{th} minute and increasing thereafter. During this time, the hit rates of our proposed system are observed to be recovering more quickly and are higher than those of the DANTE system and the random walk method. This change in the hit rate demonstrates that our proposed system effectively adapts the network topology rapidly to form a clustered network again, and so the hit rates of our proposed system are higher than those of the random walk method and the DANTE system. Thus, our proposed system can achieve effective balanced routings among the intra-group level peers through the collaborative Q-Learning method under heavy workloads, outperforms the DANTE system which evolves into a clustered network without learning the network.

Finally, the congestion rates of the peers in the entire network are evaluated, as shown in Fig.8. The congestion rates of the peers are observed to be decreasing in both the other two methods during the first 60 minutes, as there are no heavy workloads in the network. When the network suffers heavy workloads at the 60^{th} minute, the congestion rates of both the methods are observed to be increasing until the 70^{th} minute and decreasing thereafter. The congestion rates of our proposed system are observed to be recovering more quickly and are higher than those of the DANTE system and the random walk method. This is because that our proposed system not only adjusts the network topology dynamically, but also guides queries to be forwarded to those un-congested peers of the same group through the Q-Learning method.

It is interesting to observe that the characteristic changes of our proposed system are sharper under higher churns than under heavy workloads. This implies that when the peers of higher processing capacity leave the network, higher churn rates influence the dynamic characteristics of our proposed system more than the heavy workload rates. From the experiments conducted under higher churns and heavy workloads, it can be concluded that our proposed system effectively adapts to both higher churns and heavy workloads by dynamically changing the network topology through resource grouping, thereby achieving balanced routing among the intra-group level peers. Our experiment results prove that the search

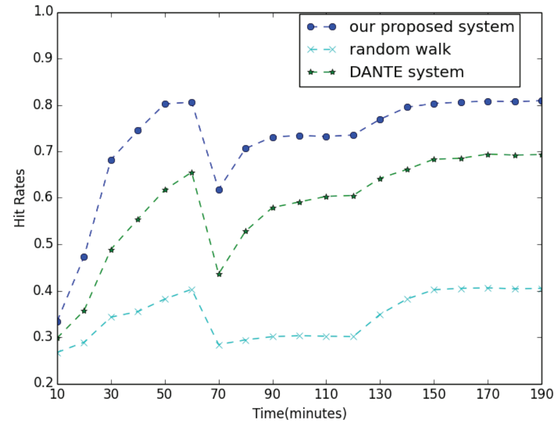


Fig. 7. Timely behavior of hit-rates of our proposed system, the DANTE system and the random walk method under heavy workloads.

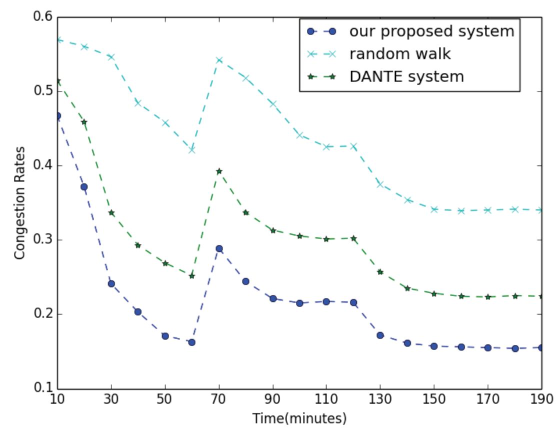
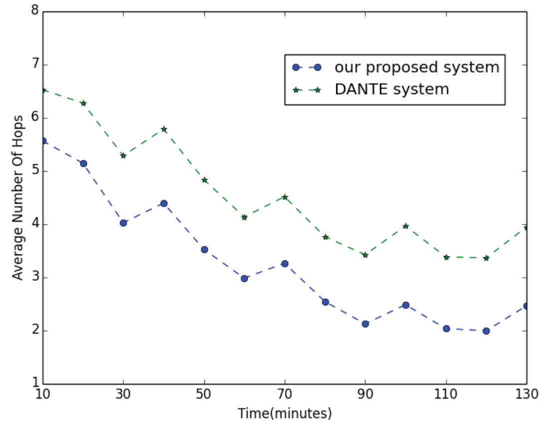


Fig. 8. Timely behavior of congestion rates of the peers of our proposed system, the DANTE system and the random walk method under heavy workloads.

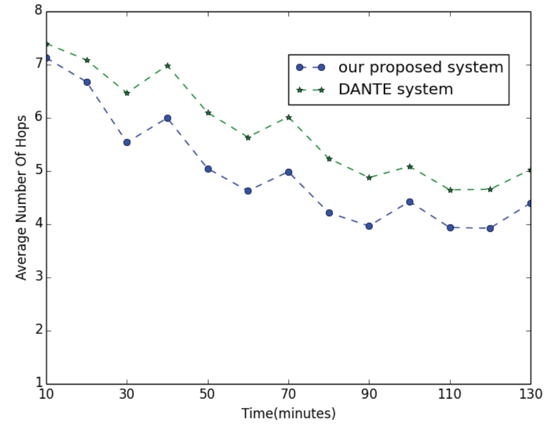
performances of our proposed system are better than those of the DANTE system and the random walk method. Also, our routing control method by collaborative Q-Learning exhibits more effective congestion control than those of the DANTE system and the random walk method.

D. Comparison with DANTE under higher churns and heavy workloads

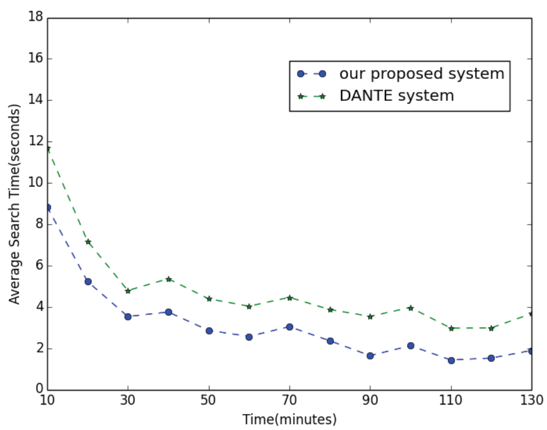
In the last subsection, we compare the efficiencies of our system with the DANTE system which is also capable of dynamically adapting the changes in the network topology. In a realistic network environment setting, both our proposed system and the DANTE system are subjected to both higher churns and heavy workloads under two different simulation scenarios. Firstly, the systems are simulated under a moderate network with a churn rate of 0.05 and tbs set to 5. In this scenario, peers enter and leave the network at a rate of 5% per 30 minutes and process $2,000$ queries per second. Secondly, the systems are simulated under an extreme network condition with a churn rate of 0.1 and the tbs set to 0.5. This extreme



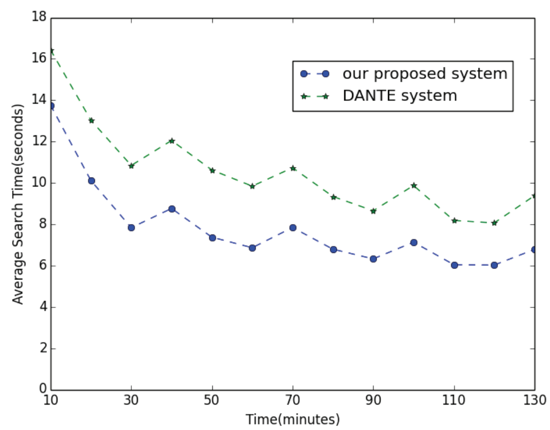
(a)



(b)



(c)



(d)

Fig. 9. Timely behavior of the network search performances of our proposed system and the DANTE system under higher churns and heavy workloads. (a)Average number of hops with a churn rate of 0.05 and $tbs = 5$, (b)Average number of hops with a churn rate of 0.1 and $tbs = 0.5$, (c)Average search time with a churn rate of 0.05 and $tbs = 5$, (d)Average search time with a churn rate of 0.1 and $tbs = 0.5$.

network scenario exhibits more dynamic characteristics with the peers leaving and entering the network at a rate of 10% per 30 minutes and process $20,000$ queries per second accordingly.

Fig.9 illustrates the timely behavior of the search performances of our proposed system and the DANTE system under the scenarios of moderate and extreme network conditions accordingly. Fig.9(a) and Fig.9(b) illustrate the timely changes of the average number of hops under the two networks conditions. Fig.9(c) and Fig.9(d) illustrate the timely changes of the average search time under the two network conditions accordingly. As shown in the four figures, both the average hops and the average search time are observed to be decreasing as time elapses. The network churn is triggered every 30 minutes under both the network conditions. Both the systems experience an increase in average hops and average search time for a shorter time after the occurrence of the churns and decreases thereafter. This demonstrates that both our proposed system and the DANTE system are capable of adapting their topologies to dynamic network environments. Also, for a given time period both the number of average hops and the average search time are higher in the extreme network than that of the moderate network. From the observed results, our proposed system performs better than the DANTE system in terms of both the average hops and the average search time, as shown in Fig.9, thereby proving the search effectiveness of our proposed system. This is because that our proposed system achieves balanced routing among both inter- and intra- group peers while the DANTE system achieves balanced routing only among the inter-group peers.

Fig.10 illustrates the success of the hit rates in the resource locating process of our proposed system and the DANTE system under the two dynamic network environments respectively. It is observed that both the two systems achieve an increase in their respective hit rates as time elapses under the two dynamic networks. Again, both the systems exhibit a decrease in the hit rates for a short time after the occurrence of the network churns and increases thereafter. This demonstrates that both the systems are capable of adapting their topologies to dynamic network environments. Though, our system exhibits a better

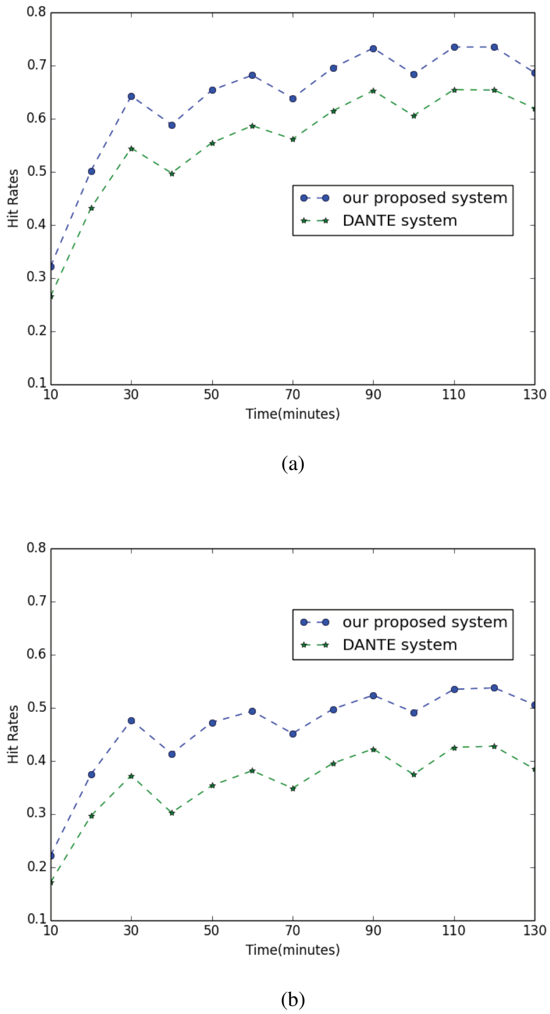


Fig. 10. Timely behavior of success of the hit rates in our proposed system and the Dante system under higher churns and heavy workloads. (a)Hit rates with a churn rate of 0.05 and $tbs = 5$, (b) Hit rates with a churn rate of 0.1 and $tbs = 0.5$.

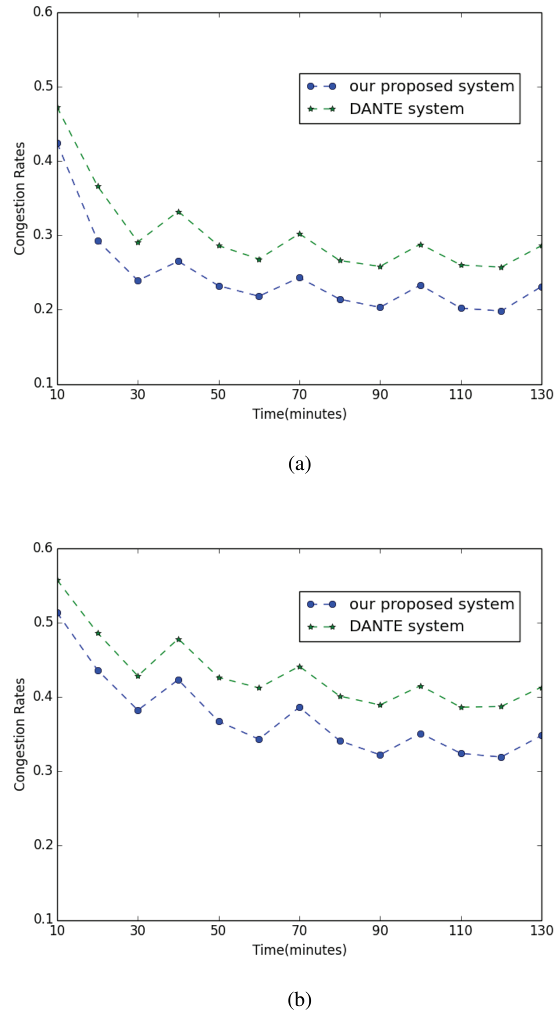


Fig. 11. Timely behavior of congestion rates of the peers of our proposed system and the DANTE system under high and heavy workloads. (a)Congestion rates of the peers with a churn rate of 0.05 and $tbs = 5$, (b)Congestion rates of the peers with a churn rate of 0.1 and $tbs = 0.5$.

performance in hit rates, as the curve of our proposed system is witnessed at 0.15 and 0.1 (in Fig.10) higher than the DANTE system for extreme and moderate networks respectively. This demonstrates that our proposed system achieves better hit rates than the DANTE system under dynamic networks, especially in extreme network conditions.

Finally, Fig.11 illustrates the timely changes in the congestion rates of the peers in our proposed system and in the DANTE system under the two dynamic network environments respectively. Congestion rates of the peers are observed to be higher in the extreme network than the moderate network for both the systems. For instance, the values of the congestion rates of the peers in our proposed system and the DANTE system in moderate network are found to be 0.21 and 0.27 at the 120^{th} minute, respectively. And the same is found to be 0.33 and 0.41 for the extreme network accordingly. Observing these values, we conclude that our proposed system achieves a lower congestion rate of the peers than that of the DANTE system. This demonstrates the efficiency of our proposed system in congestion control routing through collaborative Q-Learning.

As shown in Fig.2, where CPU time consumptions of the three methods are compared, our proposed system achieves 11.5% extra CPU time than the DANTE system and 50% extra CPU time than the random walk method respectively. This CPU time consumption is measured by simulating the three systems in stable networks. The Q-learning employed in our method causes an extra CPU time consumption whilst computing the states of the neighbor peers for avoiding congestion in the network. It is worth to add that all our simulations are conducted in a distributed environment, and so we believe that the performance of our method will not be deteriorated by the increase in the CPU time consumptions. All our experiment results demonstrate that our proposed system can achieve better search performance and exhibit better robustness and adaptability than the DANTE system, since our proposed system achieves balanced routing among both inter- and intra- group peers while the DANTE system achieves balanced routing only among the inter-group peers. Thus our proposed system can achieve efficient search performances with a moderate extra CPU time and bandwidth consumption.

V. Conclusion

P2P networks are being witnessed in many applications in the past few decades. Load balancing and decentralized resource locating approaches in such networks still suffer various limitations. In this paper, a new churn resilient system is proposed to assure alternative routing path for balancing the query loads among the peers under higher network churns. Our proposed system uses two strategies to achieve query load balancing among both inter- and intra-group peers. Firstly, a resource grouping and a rewiring mechanism is proposed to periodically cluster the peers having same resources. This strategy helps to quickly locate the requested resources in the network and enables the network to balance the query loads among the inter-group peers. Furthermore, it helps the network overlay topology to evolve from a random network into a clustered network. On the other hand, load balanced routing among the intra-group peers is achieved by employing a collaborative Q-Learning method among the peers. Our proposed collaborative Q-Learning method not only learns the network parameters such as processing capacity, number of connections and the number of resources in the peers, but also learns the congestion states of the peers. By this technique, queries are guided to avoid being forwarded to the congested peers in the network. Thus, query routings are forwarded to un-congested peers and further balanced among the intra-group level peers. Our simulation results show that the desired resources are located more quickly and query loads in the entire network are balanced by our proposed system. Also, our proposed method exhibits more robustness and adaptability under network attacks, heavy query workloads, and higher network churns than that of the random walk method and the DANTE system.

Acknowledgements

This work was funded in part by the National Natural Science Foundation of China(No.61572240,61502209,61472392), Natural Science Foundation of Jiangsu Province (BK20130528), and the Open Project Program of the National Laboratory of Pattern Recognition(NLPR)(No.201600005). Xiang-Jun Shen and Lu Liu are the corresponding authors of the paper.

We thank the reviewers for his/her thorough review and highly appreciate the comments and suggestions, which significantly contributed to improving the quality of the publication.

References

[1] J. Risson and T. Moors, "Survey of research towards robust peer-to-peer networks: search methods," *Computer networks*, vol. 50, no. 17, pp. 3485–3521, 2006.

[2] L. Liu, N. Antonopoulos, M.H. Zheng, Y.Z. Zhan, and Z.J. Ding, "A socio-ecological model for advanced service discovery in machine-to-machine communication networks," *ACM Transactions on Embedded Computing*, vol. 15, no. 2, pp. 38:1–38:26, 2016.

[3] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "Survey of research towards robust peer-to-peer networks: search methods," *Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[4] H. Chen, H. Jin, J. Sun, D. Deng, and X. Liao, "Analysis of large-scale topological properties for peer-to-peer networks," in *IEEE International Symposium on Cluster Computing and the Grid*, 2004, pp. 27–34.

[5] S.L. Mirtaheri, and M. Sharifi, "An efficient resource discovery framework for pure unstructured peer-to-peer systems," *Computer Networks*, vol. 59, pp. 213–226, 2014.

[6] L. R. Merino, A. F. Anta, L. Lópze, and V. Cholvi, "Self-managed topologies in p2p networks," *Computer Networks*, vol. 53, no. 10, pp. 1722–1736, 2009.

[7] S. Merugu, S. Srinivasan, and E. Zegura, "Adding structure to unstructured peer-to-peer networks: the use of small-world graphs," *Journal of Parallel and Distributed Computing*, vol. 65, no. 2, pp. 142–153, 2005.

[8] E. Pournaras,G. Exarchakos, and N. Antonopoulos, "Load-driven neighbourhood reconfiguration of gnutella overlay," *Computer communications*, vol. 31, no. 13, pp. 3030–3039, 2008.

[9] R. Morales and I. Gupta, "Avmon: Optimal and scalable discovery of consistent availability monitoring overlays for distributed systems," *IEEE Transactions on Parallel and Distributed Systems* , vol. 20, no. 4, pp. 446–459, 2009.

[10] H. Mashayekhi, and J. Habibi, "Combining search and trust models in unstructured peer-to-peer networks," *The Journal of Supercomputing*, vol. 53, no. 1, pp. 65–85, 2010.

[11] J. A. Torkestani, "A distributed resource discovery algorithm for P2P grids," *J. Network and Computer Applications*, vol. 35, no. 6, pp. 2028–2036, 2012.

[12] M. G. Azar, R. Munos, M. Ghavamzadaeh, and H. Kappen, "Speedy Q-Learning," in *Advances in Neural Information Processing Systems, (NIPS 24)*, 2011, pp. 2411–2419.

[13] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proc of the 16th ACM Conf. on Supercomputing*, 2002, pp. 84–95.

[14] C. Gkantsidis, M. Milena, and A. Saberi, "Random walks in peer-to-peer networks," in *Proc. IEEE INFOCOM* , 2004, pp. 120–130.

[15] A. D. Sarma, A. R. Molla, and G. Pandurangan, "Efficient Random Walk Sampling in Distributed Networks," *Journal of Parallel and Distributed Computing*, vol. 77, pp. 84–94, 2015.

[16] D. Tsoumakos, and N. Roussopoulos, "Adaptive probabilistic search for peer-to-peer networkss," in *Proc. of 3rd International Conference on Peer-to-Peer Computing*, 2003, pp. 102–109.

[17] R. L. Pereira, T. Vazäo, and R. Rodrigues, "Adaptive search radius - using hop count to reduce p2p traffic," *Computer Networks*, vol. 56, no. 2, pp. 642–660, 2012.

[18] A. Kurve, C. Griffin, D. J. Miller, and G. Kesidis, "Optimizing cluster formation in super-peer networks via local incentive design," *Peer-to-Peer Networking and Applications*, vol. 8, no. 1, pp. 1–21, 2015.

[19] J.-S. Li, and C.-H. Chao, "An efficient super-peer overlay construction and broadcasting scheme based on perfect difference graph," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 594–606, 2010.

[20] G. J. Fakasa, and B. Karakostas, "An efficient super-peer overlay construction and broadcasting scheme based on perfect difference graph," *Information and Software Technology*, vol. 46, no. 6, pp. 423–431, 2004.

[21] Z. Xu, and L. N. Bhuyan, "Effective load balancing in p2p systems," in *IEEE International Symposium on Cluster Computing and the Grid* , 2006, pp. 81–88.

[22] G. Sreenu, P. M. Dhanya, and Sabu M. Thampi, "Enhancement of bartercast using reinforcement learning to effectively manage freeriders," in *Advances in Computing and Communications*, 2011, pp. 126–136.

[23] K. Wu, C. Wu, and L. Liu, "State-based search strategy in unstructured p2p," in *Proceedings of 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, 2010, pp. 381–386.

[24] H. V. Hasselt, "Reinforcement learning in continuous state and action spaces," in *Reinforcement Learning: State-of-the-Art*, ser. Adaptation, Learning, and Optimization. Springer, 2012, vol. 12, ch. 7, pp. 207–251.

[25] J. Luo, B. Xiao, K. Bu, and S. Zhou, "Understanding and Improving Piece-Related Algorithms in the BitTorrent Protocol," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2526–2537, 2013.

[26] S. Sarolu, P. K. Gummadi, abd S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proceedings of Multimedia Computing and Networking*, 2002.