Backward-Compatible Cooperation of Heterogeneous P2P Systems

Giang Ngo Hoang, Luigi Liquori, Hung Nguyen Chan

Backward-Compatible Cooperation of Heterogeneous P2P Systems

Giang Ngo Hoang[1 2 3], Luigi Liquori[1], and Hung Nguyen Chan[4]

[1] National Institute for Research in Computer Science and Control, France
[2] Université de Nice Sophia-Antipolis, France
[3] Hanoi University of Science and Technology, Vietnam
[4] Vietnam Research Institute of Electronics, Informatics and Automation, Vietnam

Abstract. Peer-to-peer (P2P) systems are used by millions of users everyday. In many scenarios, it is desirable for the users from different P2P systems to communicate and exchange content resources with each other. This requires co-operation between the P2P systems, which is often difficult or impossible, due to the two following reasons. First, we have the lack of a dedicated routing infrastructure throughout these systems, caused by the incompatibilities in overlay networks on top of which they are built. Second, there are incompatibilities in the application protocols of these systems. In this paper, we introduce a new model for backward-compatible co-operation between heterogeneous P2P systems. The routing across systems is enabled by introducing a super-overlay formed by a small subset of peers from every system, which run an overlay protocol called OGP (*Overlay Gateway Protocol*). The incompatibilities in the application protocols are solved by a co-operation application, running on top of OGP, bridging these systems at interface level. As a real application, we present a protocol named Inter-network File-sharing Protocol (IFP), running on top of OGP, aimed at co-operation of P2P file-sharing networks. The experimental results performed on the large-scale Grid5000 platform show our model to be *efficient* and *scalable*.

1 Introduction

Nowadays, many distributed systems, such e.g. those involving peer-to-peer file sharing, peer-to-peer instant messaging, cloud computing etc. are built on top of various overlay networks. These overlay networks can differ from each other in many aspects, such as topologies, routing algorithms, types of queries, and message-encoding algorithms, and this differentiation propagates into the application protocols built on top of these overlay networks, as well. These particularities result in an *overall incompatibility* of P2P systems, and impede their cooperation. As for our motivation, there are clear advantages in facilitating the cooperation of these systems, such as increased content resources, easily achievable content redundancy, and saved storage.

Inspired by the Border Gateway Protocol (BGP) [1], we introduce a new model targeting co-operation of P2P systems. This model consists of two parts, which bridge the involved systems at the routing and application layers, respectively. The first part is the OGP routing framework, including the OGP protocol, an extension of Kademlia [2], which allows efficient routing among existing heterogeneous overlay networks. OGP is run only by a small number of peers from each of the standard overlays, in addition to their native protocols. These peers form a super-overlay (the OGP overlay) equipped with efficient algorithms to perform unicast, broadcast, and multicast of messages from

one standard overlay to the others. Peers forming the OGP overlay act as gateways for peers especially created for taking advantage of OGP which run the lightweight OGP protocol, and can reach across standard overlays they are not members of. The idea of OGP was briefly introduced in our poster paper [3]

The second part of the model is a cooperation application that makes use of the OGP routing framework, and which is responsible for bridging the P2P systems at the application layer with tasks such as transcoding the messages and data from formats of the P2P systems to intermediary formats and vice versa. Since the particular tasks of the cooperation application depends on the application domain, in this paper we only describe the principles of the cooperation application and introduce the IFP protocol for cooperation of heterogeneous P2P file-sharing systems as an example.

Our original approach ensures *backward-compatibility*, in the sense that (i) native peers can continue to operate normally, and (ii) peers that are aware of new protocols from different systems can exchange resources with each other in a transparent way. As such, the contribution of our paper is twofold: first is the introduction of a new model for cooperation between heterogeneous P2P systems consisting of a new framework for efficient inter-routing between heterogeneous overlays and principles underpinning a cooperation application for bridging these P2P systems at the application layer. Second as a concrete example of the model, we present the IFP protocol, running on top of the OGP framework, for cooperation of heterogeneous P2P file-sharing systems.

The rest of the paper is structured as follows: in Section 2, we survey the related work. Section 3 presents the system model, which was the motivation of this paper. The routing framework based on OGP and lightweight OGP protocols and the co-operation application are described in Section 4. The IFP protocol for cooperation P2P file-sharing networks is described in Section 5. Section 6 evaluates the model. Finally, in Section 7, we present our conclusions and outline future work.

2 Related work

2.1 On cooperation of P2P systems

Cooperation between P2P systems and inter-overlay routing has served as an inspiration to a number of research efforts.

In [4], the authors introduced a model for cooperation between file-sharing networks with purely *flooding-based* queries. In their model, several pairs of peers from two networks establish logical links between the two networks, and serve as bridges to transfer the search requests and the discovered files.

In [5–8], the authors deal with inter-overlay routing by using co-located nodes, i.e. nodes belonging to multiple overlays at the same time, as gateways forwarding messages between overlays. The co-located nodes also perform the transcoding of queries between overlays. In [5], the original queries from peers in one DHT are sent to the trackers that, in turn, forward them to co-located nodes, in order to reach other DHT s. In [6, 7] the messages from one DHT are forwarded to others only if they randomly touch the co-located nodes while in [8], the co-located nodes have some auto discovery mechanisms to detect each other, thus the original messages can be sent directly between them.

While the solutions in [5–7] are only for DHT s, i.e. structured overlays, the solution in [8] is for both structured and unstructured overlays. The solution in [6, 7] requires

the modification of all peers in overlays, i.e. peers which are unaware of new protocols cannot operate, which is not practical in reality. Solutions in [4,5,8] ensure the backward compatibility in the sense that the native peers which are not aware of new protocols can operate normally, which is suitable for inter-routing between existing P2P overlays.

In all previous solutions, the transcoding of messages between P2P systems is performed at the routing level, which makes these solutions become less applicative. By separating the inter-overlay routing function and application bridging function, our solution achieve more flexible and thus more applicative. This is the fundamental difference between our model and others.

The inter-overlay routing in our model is enabled via a super-overlay formed by peers running the OGP protocol, with the following main features: (i) it allows for inter-routing over *heterogeneous* overlays, including both structured and unstructured overlays; (ii) it guarantees *backward-compatibility*; (iii) it features *better control over routing*, by allowing the choice between the broadcast, multicast and unicast of the messages to all overlays, a group of overlays and a specific overlay without duplication. In previous works, where there was no control on which overlays will receive the query and, mostly, a query could reach an overlay multiple times, triggering numerous duplicated lookup processes, while not reaching some other overlays at all.

2.2 On Unicast, Broadcast and Multicast in OGP

Historically, unicast, multicast and broadcast in a DHT respectively denote the sending of a message to a peer, to a group of peers, and to all of the peers in that DHT. Two typical works dealing this issue are [9] and [10]. In OGP, we introduce new schemes of unicast, multicast and broadcast. OGP categorizes all peers belonging to one standard overlay into a group. The unicast, multicast and broadcast in OGP respectively denote the sending of a message to a group, to a number of groups, and to all of the groups in the OGP overlay. In each group, only one random node receives the message.

2.3 On Hierarchical Overlays vs. OGP

Hierarchical overlays aim at bringing a hierarchical structure into flat DHT s. In these overlays, peers are categorized into groups or netted groups, and each of these groups is a DHT. Both intra-group and inter-group routing are key-based with a unique hash function. A lookup for key k is routed to the peer closest to k. OGP, along with standard overlays can be seen as a hierarchy of heterogeneous overlays. The standard overlays can be structured or unstructured, can use different routing schemes, e.g. key-based or keyword-based, etc. The OGP overlay itself categorizes peers belonging to the same standard overlay into one group which is not a DHT. Therefore, the OGP approach does not fit the description of a hierarchical overlay.

3 System Model

In our model, there are three kind of peers:

Full OGP peers, hereafter denoted as FOGP peers, simultaneously belong to one P2P system and the OGP overlay. In addition to their native protocols, they also run the OGP protocol and the co-operation protocol. They route messages from one P2P system to the others via the OGP overlay and serve as gateways for lightweight OGP peers to reach P2P systems to which they do not belong.

Lightweight OGP peers, denoted as LOGP peers, take advantage of the inter-overlay routing provided by the OGP overlay. They belong only to one P2P system, do not participate in the OGP overlay, but keep a list of FOGP peers. In addition to their native protocols, they run the lightweight OGP protocol for communicating with FOGP peers and the co-operation protocol. LOGP peers are introduced: (i) to reach P2P systems they are not members of with low cost in terms of power processing and bandwidth, and (ii) to improve the scalability of the co-operation system by reducing the number of FOGP peers and the size of OGP overlay.

Blind peers are peers that belong to only one P2P system, are not aware of the existence of the new protocols and use only their native protocols.

3.1 Inter-routing schemes

The inter-routing algorithms are the heart of OGP protocol, including OGP *unicast*, OGP *multicast* and OGP *broadcast*. A FOGP peer can use any of these schemes. For the sake of brevity, the operation of routing a request to a random FOGP peer belonging to the destination standard overlay is hereby described as routing that request to the destination overlay. OGP *unicast* allows FOGP peers route requests into only one destination overlay different from the one the request originated from. With OGP *multicast*, a FOGP peer can selectively choose multiple destination overlays, and all of the responses are returned to the original sender. In OGP *broadcast*, all standard overlays are chosen as destination, and all of the responses are returned to the sender, just like with the multicast.

3.2 Structure of a FOGP peer

A FOGP peer, see Figure 1(a), has several components:

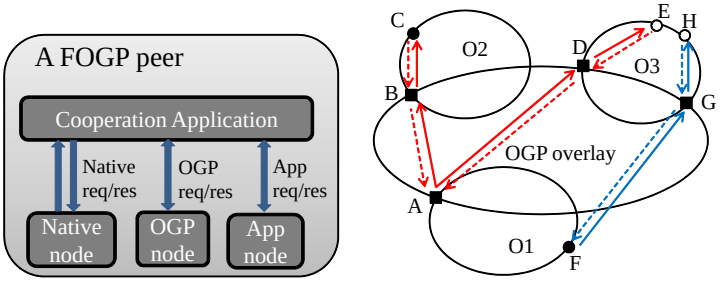

Fig. 1: (a) A FOGP peer (b) Examples of cooperation

A **Native node** participates in the P2P system to which the FOGP peer belongs, launches requests on this P2P system and returns the results to the cooperation application.

An OGP **Node** participates in the OGP overlay and provides unicast, multicast and broadcast inter-routing for the cooperation application.

An **App Node** performs tasks which are specific to a application domain.

The **Cooperation application** can launch the request on a P2P system via the Native node, on the OGP overlay via the OGP node, or ask the App node to perform certain tasks, and receive the results. It performs the transcoding of messages and data at interface level between formats of P2P systems and intermediary formats defined by itself.

3.3 Cooperation examples

In Figure 1(b), two scenarios are shown to illustrate the cooperation of three P2P systems in which a FOGP peer and a LOGP peer lookup information at overlays they are not members of. The three smaller ovals, denoted by O1, O2 and O3, represent standard overlays the P2P systems based on, while the largest oval represents the OGP overlay. The black squares A, B, D, and G represent FOGP peers, the black circles F, and C represent LOGP peers, while the white circles E, and H represent blind peers. Solid lines represent requests, while dashed lines represent responses.

First scenario. The FOGP peer A is looking for some information which is located at the LOGP peer C in overlay O2 and the blind peers E in the overlay O3. A send the request to the FOGP peer B and the FOGP peer D, belonging to O2 and O3 respectively, via its OGP node, using OGP broadcast routing. Upon receiving the request, B and D reconstruct the request to be in accordance with the possibly different format defined by the native protocols of O2 and O3 respectively, then forward it to C and E via their Native nodes. C and E then send the responses back to B and D, which reconstruct the responses to follow the format defined by cooperation protocol, and send it, along with their contact information for later communication, back to A, via their OGP nodes.

Second scenario. The LOGP peer F, belonging to overlay O1, is looking for some information located at the blind peer H in overlay O3 . It forwards, via OGP node, using OGP unicast routing, the message to G which is a FOGP peer in overlay O3. Upon receiving the message, G converts the message in accordance with the native protocol of O3, and forwards it to H via its Native node. The return path takes us back through G to F, following the native protocol of O3 first, and then the OGP protocol.

3.4 Potential Applications

Our model can used for cooperating many distributed applications, such as:

File-sharing applications. Many isolated file-sharing networks currently co-exist in the Internet, are based on various incompatible overlay protocols, and use incompatible mechanisms for downloading and uploading files [11]. By having a number of peers in each involved file-sharing network running the OGP protocol, an OGP overlay can be established to inter-connect these networks. The searching and exchanging files over networks are performed by cooperation application on top of this infrastructure. In Section 5 we develop a complete solution for cooperation of P2P file-sharing networks.

Instant messaging (IM) applications. There are many instant messaging networks with incompatible instant messaging protocols [12]. Currently, to have these networks cooperate, one can combine the many disparate protocols inside the IM client application or inside the IM server application. Our model provides another promising solution.

Cloud-based applications. Cloud systems such as Amazon EC2, or NoSql databases, such as Amazon SimpleDB [13] or Cassandra [14] usually rely on a computer cluster; the OGP framework can be used to form a routing infrastructure over the existing cloud systems while the cooperation applications on top of the OGP framework enable the exchanging data between these systems, while resolving incompatibilities.

4 System description

In this section, we describe the OGP routing framework consisting of OGP and lightweight OGP protocols and co-operation application running on top OGP framework.

4.1 OGP protocol: ID assignment

OGP identifies each standard overlay by a unique n-bit number we denote by `netID`. A FOGP peer is assigned an unique $(n+m)$-bit identifier, denoted by `ID`, consisting of two parts: the n-bit identifier of the standard overlay to which that peer belongs (`netID`), and a random m-bit number denoted by `nodeID`. Given this, and using "|" as a concatenation operator, we have that: `ID=netID | nodeID`.

4.2 OGP protocol: Routing table

A FOGP peer calculates the XOR distances, which is defined in Kademlia protocol, from itself to other FOGP peers and uses these distances to internally represent these nodes as a binary tree with the leaves of the tree are the shortest unique prefix of these distances. One important property of this binary tree is that all FOGP peers connected to the same standard overlay share a single subtree. Let the identifier of the current node be `netIDi | nodeID`. By properties of the XOR distances, we can easily see that the distance between the current node and any of the peers connected to the same overlay will share the same n-bit prefix, and, therefore, the same subtree.

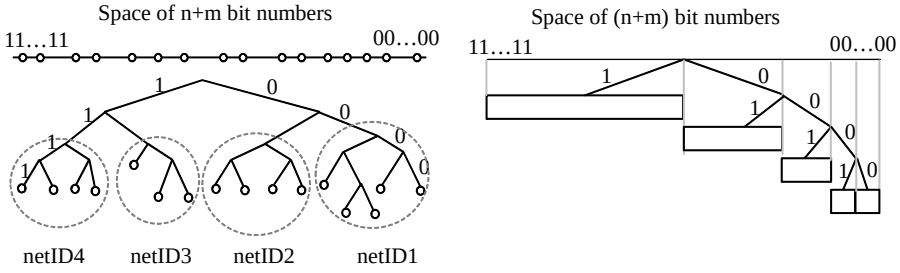

Fig. 2: (a) A binary tree of FOGP peers with n=2 (b) Routing table of a FOGP peer

Figure 2(a) illustrate the binary tree representing FOGP peers from the view of the FOGP peer whose distance metric is $00...00$, while Figure 2(b) illustrates the routing table of a FOGP peer with distance from itself is $00...000$. In the Figure, we have that n=2, i.e. `netID` is represented by 2 bits. Here, FOGP peers can belong to one of the four standard overlays, whose identifiers are `netID1`, `netID2`, `netID3`, and `netID4`. We refer to the set of all $(n+m)$-bit numbers as the *distance space*, as they represent all of the possible distances between nodes in the OGP protocol. The routing table of FOGP peer is the same as the one of Kademlia peer. A FOGP peer keeps contact information for k nodes of distance between 2^i and 2^{i+1} from itself, with $0 \leq i < (n+m)$. These lists are called k-buckets that each of which cover a range of distance space and they, together, cover the whole distance spaces. We also refer the range of distance space covered by one k-buckets as a final space.

An FOGP peer keeps a fix-sized list of other FOGP nodes, belonging to the same n-level subtree with it, for co-operation application to exploit.

4.3 OGP protocol: Routing schemes

Definitions. A n-level subtree of the OGP binary tree (Figure 2(a)) is a subtree whose prefix length equals n: all nodes connected to one standard overlay belong to a n-level subtree. From now on, by "sending a message to a subtree" we mean "sending a message to a random node belonging to the said subtree".

OGP provides three kinds of routing, namely: (i) n-level unicast, (ii) n-level multicast, and (iii) n-level broadcast which are used by a FOGP peer to, respectively, send a message to an n-level subtree, to a group of n-level subtrees, to all of the n-level subtrees that do not contain the FOGP peer. In all of the cases, each n-level subtree only receives one message. All the final subspaces, that together cover a n-level subtree with no overlap, are represented by a subspace which covers this n-level subtree.

The *Range* of a distance space $\mathtt{S}$, denoted by $\rho_{\mathtt{S}}$, is the XOR between the maximal ($\mathtt{UB}$) and minimal ($\mathtt{LB}$) numbers in $\mathtt{S}$: $\rho_{\mathtt{S}}=\mathtt{UB}\oplus\mathtt{LB}$.

The *Depth* of a distance space $\mathtt{S}$ in the routing tree of a FOGP peer, denoted by $\delta_{\mathtt{S}}$, is the number of bits of the space's prefix in that tree. From now on by "sending a message to a subspace" we mean "sending a message to a random node belonging to the said subspace". What follows are the routing schemes provided by the OGP protocol.

First routing: n-level unicast. The n-level unicast is a greedy algorithm aimed at sending a message to an n-level subtree, knowing its n-bit prefix $\mathtt{P_n}$. The sending node, which is a FOGP node and has the identifier $\mathtt{ID_0}$, first generates an m-bit random number $\mathtt{R_m}$ and concatenates it to $\mathtt{P_n}$ to form an $(n+m)$-bit identifier $\mathtt{ID}=\mathtt{P_n}\cdot 2^{\mathtt{n}}+\mathtt{R_m}$.

The initiator node then sends a $\mathtt{REPLICATE}(\mathtt{message},\mathtt{ID})$ request to the FOGP node in its routing table closest to $\mathtt{ID}'=\mathtt{ID}\oplus\mathtt{ID_0}$. Upon receiving the REPLICATE request, the recipient node checks if its identifier $\mathtt{ID_i}$ and ID have the same n-bit prefix. If so, the unicast is completed. Otherwise, the recipient node forwards the REPLICATE request to the FOGP node in its routing table closest to $\mathtt{ID}'=\mathtt{ID}\oplus\mathtt{ID_i}$. If there is no node in its routing table closer to $\mathtt{ID}'$ than itself, the recipient node drops the request.

Discussion and analysis. By this algorithm, the message jumps from one n-level subtree to an other n-level subtree to approach closer and closer to the destination n-level subtree. At each n-level subtree, the request touches only one node. Hence, we can assume that each n-level subtree is a virtual node in the overlay with n-bit identifier space. The distance of the message from the destination n-level subtree is reduced at least twice per round of request sending. Assume that the number of n-level subtrees in OGP overlay is K. After log_2K rounds of sending the request, i.e. message traverses through log_2K hops, the distance from the message to the destination subtree is $\frac{2^n}{2^{log_2K}}=\frac{2^n}{K}$. Because the n-bit prefixes of n-level subtrees are random numbers, the number of the n-level subtrees belonging to the above distance from the destination n-level subtree is 1, with high probability. That n-level subtree is the destination n-level subtree itself. Thus, it takes $O(log_2K)$ hops to reach the destination.

Second routing: n-level broadcast. This mechanism is used by a FOGP node to send a message to all n-level subtrees to which it does not belong. The main idea is that the initiator node sends the replication message to every subspaces in its routing table that does not contain the sending node and contains at least one n-level subtree. These destination subspaces, together, cover the entire distance space with no overlap. The node, receiving the message, belonging to a destination subspace, is responsible for the further broadcast of the message in this subspace, by repeating the sending operation of the initiator node, except that the entire distance space is replaced by the destination subspace. In all cases, a recipient node always excludes the subspace covers the n-level subtree containing it which already received the message from its responsible space before continuing to send the message. A node stops sending messages if the space it

is responsible for has only one n-level subtree. The entire process stops when all of the n-level subtrees have received the message. The n-level broadcast algorithm can be sketched as follows:
The initiator node sends the $\texttt{REPLICATE}(\texttt{message}, \rho_{\texttt{i}})$ request to every subspaces $\texttt{S}_{\texttt{i}}$ in its routing table which satisfy the following conditions: (i) $\delta_{\texttt{S}_{\texttt{i}}} \leq \texttt{n}$, and (ii) $\texttt{S}_{\texttt{i}}$ does not contains the initiator node, where $\rho_{\texttt{i}}$ is the range of the subspace $\texttt{S}_{\texttt{i}}$ which will receive the message.
The recipient node, i.e. the node which has received the $\texttt{REPLICATE}(\texttt{message}, \rho_{\texttt{i}})$ request, is also responsible for broadcasting the message further, to all n-level subtrees covered by subspace $\texttt{S}_{\texttt{i}}$ except the subtree it belongs to. The recipient sends the $\texttt{REPLICATE}(\texttt{message}, \rho_{\texttt{j}})$ request to every subspace $\texttt{S}_{\texttt{j}}$ in its routing table which belongs to the distance ρ_i from the recipient, and satisfies the following conditions: (i) $\delta_{\texttt{S}_{\texttt{i}}} \leq \texttt{n}$, and (ii) $\texttt{S}_{\texttt{j}}$ does not contains the recipient node ; where $\rho_{\texttt{j}}$ is the range of the subspace $\texttt{S}_{\texttt{j}}$ which will receive the message.

The above process finishes once all n-level subtrees have received the message.
Discussion and Analysis. Similar to the unicast algorithm, the message also touches only one node per subtree in broadcast scheme. Thus the n-level subtrees can be seen as virtual nodes in the n-bits overlay. The distance of the message from the destination n-level subtree is also reduced at least twice per round of request sending. Therefore, similar to unicast algorithm, n-level broadcast scheme takes $O(log_2 K)$ hops to reach the destination with high probability.
Third routing: n-level multicast mechanism. Due to lack of space, we only present the main idea of this mechanism. n-level multicast is an algorithm used by a FOGP node to send a message to a group of n-level subtrees on the OGP overlay of which it is not a member. The multicast algorithm is similar to the broadcast algorithm, with the following general idea: a node is responsible for multicasting the message within a certain distance space. To perform this task, the node divides that distance space into *multiple subspaces with no overlap*. Each subspace contains at least one n-level subtree. For each subspace that overlaps with the multicast group, if the routing table contains a contact belonging to both the subspace and the multicast group, that contact is chosen. Otherwise, the node chooses a contact belonging to that subspace which is closest to the multicast group, i.e. the node whose n-bit prefix is closest to n-bit prefix of one of subtrees belonging to the multicast group. It then sends the message to the chosen node and asks the chosen node to be responsible for multicasting the message to the n-level subtrees belonging to both that subspace and the multicast group. The above process continues until all n-level subtrees in the multicast group have received the message.
Discussion and Analysis. Using the same analysis with broadcast and unicast algorithm discussions, it takes $O(log_2 K)$ hops to multicast the message to the destination n-level subtrees with high probability.

In summary, the routing cost in three OGP routing algorithms are the same and are $O(log_2 K)$. We notice that the routing cost only depend on the number of n-level subtrees, i.e. K, and doesn't depend on the number of FOGP nodes.

4.4 Lightweight OGP protocol

The lightweight OGP protocol is performed by LOGP peers to communicate with FOGP peers. A LOGP peer maintains a routing list, which is a fixed-size list containing information about some FOGP peers in the OGP overlay by periodically asking for the

routing table of FOGP peers in its routing list and then using the information in these routing table for updating the routing list. At the bootstrap time, a LOGP peer known some bootstrap FOGP peers via external mechanisms such as from websites. A LOGP peer sends messages to standard overlays of which it is not member by simply sending these messages to the first FOGP peer in its routing list which will forward its messages.

4.5 Cooperation application

The cooperation application in a FOGP peer is built on top of the OGP routing layer, and is responsible for following tasks: (i) launching the delivery of requests to P2P systems via the Native node or the OGP node or both and receiving results from these nodes, and (ii) transcoding of messages and data between formats of P2P systems and the intermediary formats defined by itself at interface level, and (iii) communication with each other, via the App node.

The first and the third tasks can be achieved easily. The intermediary formats, in the second task, vary from application to application. Therefore, we cannot introduce a common intermediary formats. As a case study, we show in the next section the IFP protocol, running on the top of OGP, which is an application allowing heterogeneous P2P file-sharing networks to cooperate, together with the respective intermediary formats.

5 Case Study: cooperation of P2P file sharing networks

We introduce the IFP protocol for cooperation between heterogeneous P2P file sharing networks. The IFP constitutes two schemes of cooperation, namely inter-network downloading and inter-network uploading, allowing users to download files from and upload files to P2P file-sharing networks, respectively.

The IFP protocol is responsible for the following tasks: (i) launching the processes of searching, downloading and uploading files on P2P file-sharing network contains the peer and receiving the results via the Native node, (ii) launching the delivering of search requests or upload requests to P2P file-sharing networks don't contain the peer and receiving the results from these networks via the OGP node, (iii) transcoding of search requests, search results, download requests and upload requests between the formats defined by P2P file-sharing networks and the intermediary formats defined by IFP, (iv) delivery of download requests on P2P file-sharing networks don't contain the peer and exchanging files via the App Node, and (v) communicating with, and transferring the files between FOGP nodes, via the App node.

5.1 Inter-network downloading

Transcoding of messages. IFP defines its own formats for the search request, the search result and the download request. The transcoding of these messages between IFP format and formats of P2P file-sharing networks happens at the FOGP gateways.

Most of P2P file-sharing networks have search capability with keyword search. The search criteria can include file attributes. One exception is BitTorrent, the most widely used P2P file-sharing network, which does not have search capability. However, BitTorrent users can still search the torrent files from websites using keywords. Therefore, IFP defines its search request containing keywords and file attributes; the search result contains the notification of no search capability in case of BitTorrent or the list of matched files along with attributes in other cases; the download request contains the torrent file in case the destination network is BitTorrent and information of the expected file in other cases. The three messages are illustrated in Figure 3.

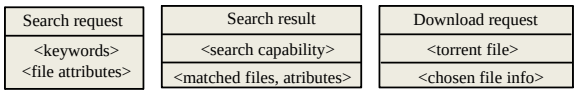

Fig. 3: Formats of search messages defined by IFP

Algorithm. The IFP protocol functions as follows:
Step 1: The *initiator* peer sends the search request to destination networks via its OGP node. The case that the *initiator* search files on its network is trivial, thus is not shown.
Step 2: A *recipient* peer, which is a FOGP peer belonging to the destination network, upon receiving the search request, acts as follows: if the destination network is BitTorrent, the *recipient* return BitTorrent indication i.e. no search capability. Otherwise, the *recipient* converts the search request from IFP format to the format defined by the destination network. It then launches the search on this network via its Native node. Upon receiving the search result, the *recipient* convert the this result to the IFP format and then sends the result along with its information to the *initiator* via its OGP node.
Step 3: Upon receiving the search result from the *recipient* peer, if the result indicates the destination network as BitTorrent, then the user search and download the torrent file from a website, and directly send the torrent file to the *recipient* in the download request via its App node. Otherwise, if the sought file exists on the destination network, the *initiator* peer directly contacts the *recipient* asking it to retrieve the file via the App node.
Step 4: The *recipient* peer, upon receiving the download request, retrieves the list of peers hosting the file via its Native node. If the destination network supports multiple-source download, the *recipient* peer can, via its App node, ask some other FOGP peer belonging to destination network, which are in its FOGP peer list, to download some parts of the file. Otherwise, it is responsible for downloading the entire file using its Native node.
Step 5: The FOGP peer, upon receiving the request for downloading some parts of the file, downloads these parts via its Native node.
Step 6: Upon receiving the file or file parts from the hosting peers after issuing the download request, the recipients send the file or file parts back to the *initiator* node via App nodes and the information for joining the parts is sent along with these parts.

5.2 Inter-network uploading

The inter-network uploading scheme allows the users to upload their files to any network. The processes of inter-network uploading is as follows:
Step 1: The *initiator*, which is a FOGP or a LOGP peer sends the upload request to *recipients* which are FOGP peers belonging to a group of networks that the *initiator* wants to replicate the file to, via its OGP node.
Step 2: Upon receiving the request, a *recipient* sends the response notifying the *initiator* whether the upload request is accepted or not via its OGP node,
Step 3: Upon receiving the notification, if the upload request is accepted, the *initiator* peer sends the file to the *recipient* via its App node.
Step 4: Upon receiving the file, if the recipient's network is BitTorrent, the *recipient* creates a torrent file for the file and registers the torrent file with some trackers, using its Native node. Then the *recipient* sends the torrent file back to the *initiator* via the

App node. If the *recipient*'s network isn't BitTorrent, it uploads the file to its standard network using the Native node and send the acknowledgement back to the *initiator* via the App node.

6 Evaluation

We first evaluate the OGP routing framework in three following aspects: routing efficiency in terms of ratio of successful inter-overlay routing; routing cost, i.e. number of hops on OGP overlay that successful routings have traversed, and traffic generated by OGP and lightweight OGP protocol. Then, we evaluate the efficiency of cooperation of P2P file-sharing networks in term of ratio of successful download file and upload file operations between networks.

6.1 Metrics

OGP framework. The routing efficiency is characterized by the metrics R^{fogp} R^{fogp}_{lookup}, R^{logp} and R^{logp}_{lookup} defined as follows. R^{fogp} is the success ratio for requests sent from a FOGP peer to the standard overlay containing the requested data and then back to the originator. R^{logp} is success ratio of request sent to a FOGP peer from a LOGP peer and the corresponding response is turned back. R^{fogp}_{lookup} and R^{logp}_{lookup} are the success ratios of inter-overlay lookups initiated by FOGP and LOGP peers, respectively.

The routing cost is represented by P^{fogp} metric which is the number of hops on OGP overlay that a request passed in a successful routing. The bandwidth generated by OGP and lightweight OGP protocols in a FOGP peer and a LOGP peer respectively during one minute are denoted by T^{fogp} and T^{logp}.

Cooperation of P2P file-sharing networks. The cooperation efficiency is characterized by the metrics $R^{fogp}_{download}$, $R^{logp}_{download}$, R^{fogp}_{upload} and R^{logp}_{upload} defined as follows : $R^{fogp}_{download}$ and $R^{logp}_{download}$ are the ratios of a FOGP peer and a LOGP peer, respectively, successfully download a file which does not exist in the peer's network but exists in other networks. The two metrics, R^{fogp}_{upload} and R^{logp}_{upload}, are the ratios of a FOGP peer and a LOGP peer, respectively, successfully upload their files to networks of which they are not members.

6.2 Setup

To evaluate the OGP framework, a complete system, in which the OGP overlay is used to interconnect twenty 50-node networks of three types Kademlia, Chord [15] and Gnutella [16] has been deployed. The experiments consisted in testing the lookup of random data distributed across all of the standard overlays, with each piece of data unique. The FOGP and LOGP peers periodically looked up a random piece of data on any of the standard overlays of which they are not members.

To evaluate the cooperation of P2P file-sharing networks, we deployed a complete system in which OGP, lightweight OGP and IFP protocols are used to cooperating three P2P file-sharing networks: BitTorrent, Gnutella-based and Kademlia-based which represent for three typical kinds of P2P file-sharing networks currently: (i) the network without search capability, (ii) the network with the flooding search and (iii) the network with DHT search, respectively. The FOGP peers and LOGP peers periodically download/upload random files from/to the networks of which they are not members.

The experimental platform is the French Grid5000, which aims at providing a nation-wide testbed to study large scale parallel or distributed systems.

All experiments are performed in churn condition with the lifetime mean of nodes is set to 3600 seconds and following the Pareto distribution. Each experiment includes 3 successive phases: *1: initial phase*, *2:stabilizing phase* and *3:evaluation phase* in which *1:*nodes are created and join overlays; *2:*the system becomes stable; *3:*the statistics are collected. The duration of each of two last phases is T with T is the lifetime mean of a node in that experiment. Each experiment is run 5 times. Average values and corresponding standard deviations of the metrics are plotted in the figures. The parameters of experiments are illustrated in the Table 1.

Experimental parameters	Evaluation 1		Evaluation 2	
	Scenario 1	Scenario 2	Scenario 1	Scenario 2
% of FOGP peers	6, 10, 20, 30	10, 20	3, 6, 10, 20, 30	6, 10
% of LOGP peers	0	10, 20, 40, 60	0	10, 20, 40, 60
Type of networks	Kademlia, Chord, Gnutella		BitTorrent, Kademlia-based, Gnutella	
No. of networks	20		3	
No. of nodes per network	50		100	
Lifetime mean (second)	3600			

Table 1: Values of experimental parameters

6.3 Experiment results: Efficiency

This section evaluates the routing efficiency of OGP framework which is characterized by R^{fogp}, R^{fogp}_{lookup}, R^{logp} and R^{logp}_{lookup} metrics and the efficiency of cooperating P2P file-sharing networks, represented by $R^{fogp}_{download}$, $R^{logp}_{download}$, R^{fogp}_{upload} and R^{logp}_{upload}. The values of the metrics for FOGP and LOGP peers are illustrated in the Figure 4 and in the Figure 5.

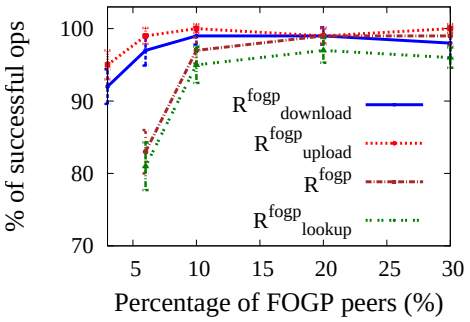

Fig. 4: Operation efficiency of FOGP peer

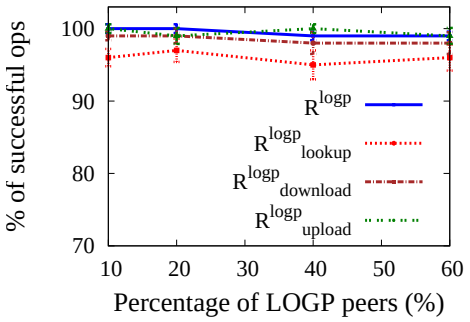

Fig. 5: Operation efficiency of LOGP peer

Figure 4 shows the four lines: $R^{fogp}_{download}$, R^{fogp}_{upload}, R^{fogp}_{lookup} and R^{fogp} share mostly the same trend. The two lines R^{fogp} and R^{fogp}_{lookup}, dramatically increase from 83% to 97% and from 81% to 95% then slightly vary in the range from 97% to 99% and from 95% to 97 % as the percentage of FOGP peer increase from 6% to 10% and then to 30%. Similarly, the two lines $R^{fogp}_{download}$ and R^{fogp}_{upload}, come from 92 % to 97% and from 95% to 99%; then slightly vary in the range from 97% to 99% and from 99% to 100% as the

percentage of FOGP peer increase from 3% to 6% and then to 30%. The R^{fogp}_{upload} line mostly stays above $R^{fogp}_{download}$ line and the two lines $R^{fogp}_{download}$ and R^{fogp}_{upload} mostly stay above the R^{fogp}_{lookup} line.

Analysis and discussion. The OGP protocol achieves routing efficiency in the interconnecting system of 20 overlays with only a small percentage of FOGP, namely 10%, while the cooperation of 3 file-sharing network achieve efficiency with an even smaller percentage of FOGP peers of 6%. In the first evaluation, the R^{fogp} and R^{fogp}_{lookup} are not less than 97% and 95% while in the second, $R^{fogp}_{download}$ and R^{fogp}_{upload} are not less than 97% and 99%.

The reason for these results is as follow: with 6% of FOGP peers in the first evaluation and with 3% of FOGP peers in the second one, the number of FOGP peers per overlay is 3 in both evaluations, meaning that there are 3 gateways to enter each standard overlay. In a churn environment, some gateways can go down for at certain time. During this time, some other FOGP peers do not have any backup gateways for the downed gateways in their routing tables, as the number of gateways to enter an standard overlay is only 3. With 10% of FOGP peers in the first evaluation, and 6% in the second one, there are 5 and 6 gateways to enter a standard overlay, respectively, and these numbers appears to be sufficient for the FOGP peers to build their routing table with quite enough backup. The R^{fogp}_{upload} line stays above $R^{fogp}_{download}$ line because the number of communication in the inter-network upload operation is smaller than the one in inter-network download operation while each communication has a probability of failure in the churn environment. Same arguments lead the $R^{fogp}_{download}$ and R^{fogp}_{upload} lines stay above the R^{fogp}_{lookup} line.

The Figure 5 shows that, with the percentage of FOGP peers is set to 10%, the two lines R^{logp} and R^{logp}_{lookup} vary from 99% to 100% and from 95% to 97% when the percentage of LOGP peers comes from 10% to 60%. On the other hand, the $R^{logp}_{download}$ and R^{logp}_{upload} values slightly vary in the range from 98% to 99% and from 99% to 100% respectively with percentage of FOGP peers is set to 6%. The experimental results in the cases that the percentage of FOGP peers is set to 20% in the first evaluation and 10% in the second one are similar to those in the illustrated cases that the percentage of FOGP peers is set to 10% and 6% respectively, thus are not shown in the figure for the sake of clarity.

Analysis and discussion. The experiments shows an important results. The LOGP protocol achieves highly routing efficiency, namely R^{logp} is nearly 100% for all percentage of LOGP. Because the LOGP peers rely on FOGP peers for inter-overlay cooperation, this means that LOGP peers perform the inter-overlay operations with the efficiency nearly the same as the efficiency of FOGP peers.

6.4 Routing cost

Figure 6 shows values of the P^{fogp} metric, i.e. the number of hops on OGP overlay that a request passed in a successful routing in the first evaluation, increase from 3.9 to 4.2 and then slightly vary in the range from 4.2 to 4.4 when the percentage of FOGP peers increase from 5% to 10% and then to 30% respectively.

Analysis and discussion. The experiment results confirm the evaluation of routing cost on OGP overlay. In our experiments, K=20, thus the expected value of P^{fogp} is $O(log_2 20)$

or approximately 4.3 hops, i.e. a constant. The experiment shown that the values of P^{fogp} is approximately the expected constant when the percentage of FOGP nodes is larger than 10% while smaller than expected constant with the 6% of FOGP. The reason is the following: in churn environment, the routing with more hops fails at higher probability than the routing with less hops (each hop has a certain probability of failure). In our experiment, at 6% of FOGP peers, the ratios of success routing, i.e. R^{fogp}, are only 83%. This means the number of routing with more hops which fails is considerably higher than the number of routing with less hops which fails. Hence the average hops of success routing, i.e. P^{fogp}, is lower than the expected constant.

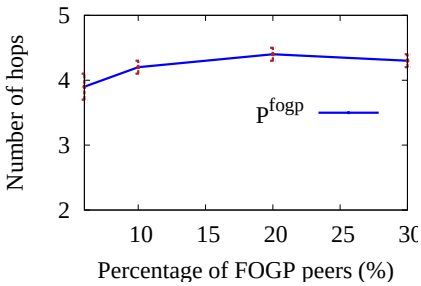

Fig. 6: The OGP routing cost

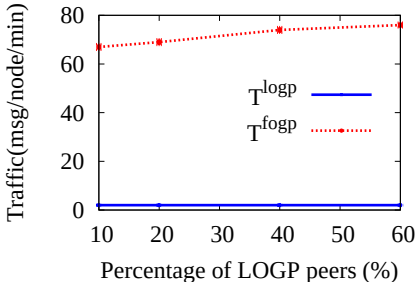

Fig. 7: Traffic generated by a peer

6.5 Generated traffic

In Figure 7, the two lines T^{fogp} and T^{logp}, respectively, represent the traffic generated by OGP and LOGP protocols in a peer during one minute while the percentage of LOGP peers increase from 10% to 60% and the percentage of FOGP peer is set to 10%. As the percentage of LOGP peers increase from 10% to 60%, the T^{fogp} increase from 67 to 76 messages/node/minute while the T^{logp} is a horizontal line at the traffic of 2 messages/node/minute.

Analysis and discussion. The experiment results are meaningful. The LOGP protocol generates little traffic (2 messages/node/minute), which also does not depend on the percentage of LOGP peers in the lookup system. On the other hand, traffic generated by a FOGP peer increases only 13% as the percentage of LOGP peers increases from 10% to 60%. These results show that our model is scalable in terms of generated traffic.

7 Conclusions and Future Work

In this paper, we have introduced an efficient model for backward-compatible co-operation of heterogeneous P2P systems. The model consists of the OGP framework for inter-overlay routing and the co-operation application on top of the OGP framework, mapping the interface of these P2P systems to a mediatory interface. We also introduce the IFP protocol, which, along with OGP framework, enables the co-operation of heterogenous P2P file-sharing networks.

The evaluations show that having a small number of FCFS peers, namely not less than about 5 FOGP peers per network, is sufficient for achieving routing efficiency in 20 inter-connected overlays and achieving efficiency in the co-operation of 3 different P2P file-sharing networks. The experiments confirm that the routing cost on the OGP overlay is logarithmic to the number of overlays inter-connected by the OGP overlay.

We also notice that the LOGP peers need only one hop to reach FOGP peers. The experiments also show that the traffic generated by a FOGP peer increases only 13% as the percentage of LOGP peer rises from 10% to 60%, while a LOGP peer generates the traffic nearly as same as that generated by a blind peer (only 2 messages larger than the blind peer). These, coupled with control over the routing between standard overlays, make our model scalable. The experiment results show that the LOGP peers achieve routing efficiency is nearly the same as the FOGP peer, namely R^{logp} is not less than 99%. As a matter of fact, we can see that the LOGP peers achieve nearly the same routing efficiency and co-operation efficiency, while paying a small cost.

Our further work on this topic is a solution aimed towards a real-world P2P file-sharing network and a model for co-operation of P2P instant messaging networks.

Acknowledgements. The authors are grateful to Petar Maksimović for a careful reading of the paper and Vincenzo Ciancaglini for the useful discussions.

References

1. Y. Rekhter, T.L.: A border gateway protocol 4 (bgp-4). http://tools.ietf.org/html/rfc4271.
2. Maymounkov, P., Mazières, D.: Kademlia: A Peer-to-peer Information System Based on the XOR Metric. In: Proc. of IPTPS. (2002).
3. Ngo, G., Liquori, L., Ciancaglini, V., Maksimovic, P., Chan, H.: A backward-compatible protocol for inter-routing over heterogeneous overlay networks. In: Proc. of SAC. (2013).
4. Konishi, J., Wakamiya, N., Murata, M.: Proposal and evaluation of a cooperative mechanism for pure p2p file sharing networks. In: Proc. of BioADIT. (2006).
5. Cheng, L.: Bridging distributed hash tables in wireless ad-hoc networks. In: GLOBECOM. (2007).
6. Liquori, L., Tedeschi, C., Bongiovanni, F.: Babelchord: a social tower of dht-based overlay networks. In: ISCC. (2009).
7. Liquori, L., Tedeschi, C., Vanni, L., Bongiovanni, F., Ciancaglini, V., Marinkovic, B.: Synapse: A scalable protocol for interconnecting heterogeneous overlay networks. In: NETWORKING. (2010).
8. Ciancaglini, V., Liquori, L., Ngo, G., Maksimovic, P.: An extension and cooperation mechanism for heterogeneous overlay networks. In: Networking Workshops. (2012).
9. Castro, M., Druschel, P., Kermarrec, A.M., Rowstron, A.I.: Scribe: a large-scale and decentralized application-level multicast infrastructure. IEEE J.Sel. A. Commun. (2006).
10. El-Ansary, S., Alima, L., Brand, P., Haridi, S.: Efficient broadcast in structured p2p networks. In: IPTPS. (2003)
11. Wikipedia: Comparison of file sharing applications. http://en.wikipedia.org/wiki/Comparison_of_file_sharing_applications.
12. Wikipedia: Comparison of instant messaging protocols. http://en.wikipedia.org/wiki/Comparison_of_instant_messaging_protocols.
13. Amazon: simpledb. http://aws.amazon.com/simpledb/.
14. The Apache Cassandra project. http://cassandra.apache.org/.
15. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: SIGCOMM. (2001).
16. Wikipedia: Gnutella. http://en.wikipedia.org/wiki/Gnutella.