Proceedings of the 9th International Conference on Applied Informatics
Eger, Hungary, January 29–February 1, 2014. Vol. 1. pp. 171–180
doi: 10.14794/ICAI.9.2014.1.171

Exposing metadata in BitTorrent as Linked Open Data

Rita Fleiner, Attila Györök

Óbuda University, Budapest
fleiner.rita@nik.uni-obuda.hu, attilagyorok@gmail.com

Abstract

The process of obtaining a digital resource in a P2P network consists of three main steps: (1) the identification of the resource using some meta-data information, (2) the identification of the source where the specific content can be downloaded from and (3) the download of the content. Studying BitTorrent as the most successful open Internet application for content distribution one can find that although search is a key part of the its infrastructure, the possibility for meta-data search is completely missing from the system. In this publication we suggest a framework for exposing metadata of available content in BitTorrent as Linked Open Data. This makes it possible to formulate complex queries about the available content for the download. Furthermore we present the design of the concept and a suggestion for the ontology.

Keywords: BitTorrent protocol, Linked Open Data, Metadata

MSC: 68M14

1. Introduction

In our days there are quite a few well-functioning methods for content distribution in the web supported by properly-designed infrastructures. But there exist digital resources without concrete data owners that are open and should be available for anyone. In many cases open data is made available by governmental subsidies or private donations with financially restricted opportunities. The main goal of this publication is to describe a new framework for the sharing of open data providing possibility for sophisticated metadata search on the available content where content distribution is based on existing P2P technologies. According to [1] P2P content distribution process is composed of the following three steps:

1. WHAT phase: Identify which file you want from some meta-data criteria

2. WHERE phase: Work out where it is (potentially in multiple locations or pieces)
3. HOW phase: Download it (from one or multiple locations)

Our goal was to work out a model that achieves the above described three tasks, namely the search of the content, the determination of its location and the process of the download relying as much as possible on existing technologies. We chose to use BitTorrent protocol for the download phase, but noticed a need for a well-designed search phase. The objectives of the paper is to explore the currently available P2P content distribution protocols, to study the various search schemes in the current P2P landscape, to set up requirements for the search process and finally to propose a search scheme that fits the specified requirements.

2. P2P content distribution

Peer-to-peer networks. A peer-to-peer (P2P) network is a type of decentralized and distributed network architecture in which individual nodes in the network called peers act as both suppliers and consumers of resources. Peer-to-peer networks implement a virtual overlay network on top of the physical network topology, where the nodes in the overlay form a subset of the nodes in the physical network. Peer-to-peer overlay networks in practice possess various degrees of centralization. We have to mention that all systems at least to a limited extent always use some central administration server e.g. for initial system bootstrapping or for allowing new users to join the network by providing the access of the current users.

According to the categorization in [2] we identify the following three main types of architectures. In the Purely Decentralized Architecture all nodes perform exactly the same tasks, acting both as servers and clients, and there is no central coordination of their activities. The original Gnutella network [3] is an example of this type. In the Partially Centralized Architecture some of the nodes assume a more important role, acting as local central indexes for data shared by local peers. The more recent version of the Gnutella protocol [4], the Kazaa system [4] and Edutella [5] are examples of this type. In the Hybrid Decentralized Architecture there is a central server maintaining metadata information about the shared content stored by the nodes. The end-to-end interaction and data exchange take place directly between the peers. The central servers facilitate this interaction by providing search facilities for the content and its location. BitTorrent [6] is an example for this type.

By studying the network structure behind P2P systems we can distinguish two different cases. In the unstructured case the overlay network is created ad hoc as nodes and content are added. Searching mechanism for a specific content can be a brute force method, like flooding the network with the queries. Or search can be based on more sophisticated strategy like random walks or routing indices. Systems, like Napster and Gnutella fall into this category. In the structured case the creation of the overlay network is based on some specific rules. These systems provide a mapping between the content identifier and the location of the content in the form of a distributed routing table. An example for structured system is Chord [7], which is a peer-to-peer routing and location infrastructure that performs a mapping of content identifiers onto node identifiers. In Chord data items and nodes are identified by keys. The keys are assigned to the content and to the nodes by a deterministic consistent hash function. All node identifiers are ordered in an "identifier circle" modulo 2m. The (key k, data item) pair is stored at a node whose identifier is equal to, or follows k, in the identifier space. This node is called the successor node of key k. The use of consistent hashing tends to balance load, as each node receives roughly the same number of keys.

BitTorrent protocol. We chose to base our content distribution framework on BitTorrent protocol, which is in our days the most successful open Internet application for content distribution. BitTorrent is very effective in distributing large files, including open-source software distributions. BitTorrent software is free and many clients' versions are open source. BitTorrent is successful, because it is efficient, open, easy to deploy and free [8], [9].

In BitTorrent content is divided into many pieces. A single peer is able to download many fragments simultaneously and it does not need the whole resource to share it with other peers. Peers sharing the same resource form a P2P network, called a torrent or a swarm. At any given instant of time, each peer in a torrent is either a leech or a seed; a seed possesses the entire file, where as a leech possesses only a portion of the file.

BitTorrent Ecosystem consists of millions of BitTorrent peers, hundreds of active trackers, and dozens of torrent discovery sites. Users search for a specific content at a torrent-discovery site, such as Pirate Bay. For years, BitTorrent protocol used .torrent files to store information on the shared content. These files are stored by torrent-discovery sites and hold several types of data, like names of the files, the hash codes of the files and the URL of the tracker site. Trackers are responsible for controlling the resource transfer between the peers. A user learns the URL of the tracker from the .torrent file. Then the user asks the tracker for the locations of the peers having the specific content. After getting answer from the tracker, the user can start to download the desired content.

Recently there is a trend to replace .torrent files with Magnet links. They are just URI's, there is no files associated with them. These links contain several parameters, like the hash value of the torrent and links to trackers used by the torrent. Probably magnet links will eventually replace .torrent files. For example in February 2012, The Pirate Bay started to use magnet links exclusively. Figure 1 shows the content download process in BitTorrent.

Nowadays the role of the tracker is decreasing. Several of the most popular client types (like uTorrent and Vuze) support distributed tracker services, such as DHT (Distributed Hash Table) and PEX (Peer Exchange) [10]. In this case the tracker is used mostly to initiate the connection with a swarm. After the connection is established, BitTorrent extensions enable the communication between peers

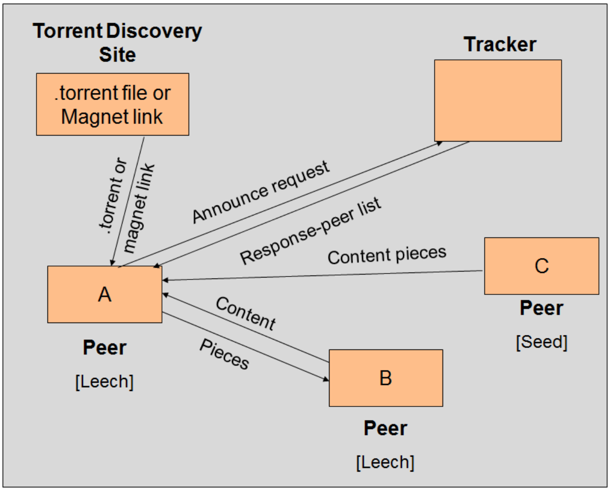

Figure 1: Content download process in BitTorrent

without using a tracker as part of the swarm. The problem is that different BitTorrent client types employ different, with each other often incompatible protocols for DHT and PEX, thus achieving the splitting of torrents for smaller pieces.

3. Search schemes in the P2P landscape

In this section we study the various search schemes in the current P2P landscape. Regarding the architecture of the search three main categories can be identified: centralized search, hash-based index distribution, and query broadcast. Centralized search is a three step process. The search engine first identifies and aggregates all the documents through an automated process (like through web crawlers). It then inserts the content of the documents into an index. It is a data structure optimized for efficient search. Finally, in case of a query, the search engine retrieves from the index the set of documents containing the query term. In this scheme centralized index is built for document discovery.

In case of hash-based index distribution there is a hash function that returns the node storing the data with a given ID. To locate a data object, we have to hash the object's name to obtain a key that map to a unique node in the network. Finally the query message has to be routed to the node handling the object.

In the broadcast approach, each node searches its local index for a given query and then forwards the query to all of its neighbors. After some time nodes doesn't broadcast the query further. Each node passes its hits back to the initial requester who then selects which documents to download. [11]

A search for a specific content can be based either on full-text indexing achieving a keyword search, or on some metadata of the content. The most common way to implement keyword search in information systems is by inverted index. An inverted index is a set of pairs (w,O), where w is a keyword, and O is the set of objects containing this keyword. Once an inverted index is built, a set of keywords can be entered to find all objects that contain these keywords. To implement keyword search in a P2P network, a distributed version of inverted index can be built. A given keyword is used as a key to determine the node that is responsible for the keyword, and obtains objects that contain the keyword. By taking a join operation, one can retrieve objects with a given keyword set. [12]

Metadata represents basic information about data, which can be used to find and work with particular instances of data easier. Because in content distribution, besides documents there are several other types of resources, like images, videos, sounds, spreadsheets, where obviously full text indexing is not an option, we decided to look for a search scheme which is based on metadata. Metadata publishing is the process of making metadata data elements available to external users.

In the present BitTorrent structure metadata about the available content can be found at discovery sites, torrent files, in Magnet links and in DHTs. In BitTorrent currently used metadata is for example torrent category, torrent upload time, torrent uploader, the number of downloads, the torrent infohash, creation time, the list of trackers, data file size. Presently the structure and the format of metadata in BitTorrent are not standardized and the supported search type is keyword search on the metadata content.

In the BitTorrent infrastructure search for a desired content plays a key role. If users cannot find content, BitTorrent's capability to share the content is useless. BitTorrent protocol does not contain the way how metadata about the available content should be organized and distributed. The user must use other mechanisms - such as web based torrent discovery sites for the search process. Torrent discovery sites operate as web-based torrent search engines and indexes providing centralized search operations for users. They keep a list of active.torrent files or Magnet links and they contain limited metadata about the distributed content. The torrent-discovery sites use different formats for presenting torrent meta-data information.

Torrent discovery sites are often targets for legal attacks from content owners. Research shows [8] that if a torrent-discovery site was shut down, only a small percentage of the active torrents would no longer be indexed by other sites. It implies that the removal of a torrent-discovery site would not greatly impact the BitTorrent Ecosystem.

4. Proposed metadata scheme for BitTorrent

Metadata publishing is the process of making metadata data elements available to external users. In order to achieve a proper metadata design, the requirements towards metadata registry and search have to be specified and metadata registry and metadata format have to be chosen.

We found that sophisticated search in open data distribution can be achieved only by centrally managed metadata registry. In BitTorrent architecture the natural place for it is the torrent discovery site. Our aim was to base the format and methods on existing standards. We want to provide selective access to metadata, which means that complex queries should be supported. Furthermore we want to provide the possibility to link metadata elements of one registry to information resources of other sources. This implies portability and semantic interoperability between different metadata resources should be supported. To fulfill these requirements we suggest publishing metadata in BitTorrent as Linked Open Data (LOD).

Linked Open Data. The Linked Open Data concept was invented by Tim Berners-Lee in 2006, who outlined four principles for Linked Data [13]. These principles are related to publishing and interlinking structured data on the Web in such a way that it can be read automatically by computers. This method enables data from different sources to be connected and queried. The four principles by Tim Berners-Lee are the following. The first rule says that things should be identified by URIs. In the context of BitTorrent things are the available contents, so each content needs an URI identifier. According to the second rule, URIs that identify resources should be resolvable HTTP URIs. The third Linked Data rule proposes to deliver useful information whenever a URI is dereferenced. In case of BitTorrent by dereferencing URIs clients should get back metadata information about the related content. BitTorrent Discovery Site should support HTML and some other RDF serialization formats, like RDF/XML, Turtle or Notation3. The server should use content negotiation to decide which representation to deliver. The forth Linked Data rule recommends that metadata records should contain links to other related resources, in BitTorrent context this could be for example the website dbpedia.org. The standard data model for Linked Open Data is the Resource Description Framework (RDF). In RDF data is structured in triples in the form of subject, predicate and object, which is called a statement. The predicate specifies how the subject and object are related. The subject and the predicate are both URIs and the object is a URI or a string literal. SPARQL is an RDF query language, designed to retrieve and manipulate data stored in RDF format.

By publishing BitTorrent metadata in LOD we mean that metadata source would consist of RDF triples at Torrent Discovery Sites. At the Torrent discovery site the metadata will be uploaded by the users to the server in a controlled way (e.g. through a form supported by the ontology).

Ontology. A crucial concept in Linked Open Data is ontology. Ontology formally represents the set of concepts within a domain, it describes the classes of the entities, their properties and the relationships among the classes. A similar concept to ontology is vocabulary. Vocabularies define the concepts, their relationships and constraints that are used to describe and represent an area of concern. The role of vocabularies and ontologies are to help data integration when ambiguities may exist on the terms that are used in the different datasets, or when a bit of extra knowledge may lead to the discovery of new relationships.

There is no clear division between the terms of vocabulary and ontology and often these two terms are used interchangeably. For example Dublin Core [14] is referred in some cases as ontology and in other cases as vocabulary. We suggest to use the word ontology for more complex collection of entities and classes containing them, and to use the term vocabulary when there is a simple relationship among the database entities and their classes. If there are more classes defined in the schema and there is a hierarchy among the classes, then it is appropriate to use the term ontology instead of vocabulary.

In the process of creating the schema for the Linked Open Dataset it is advisable to reuse as much as possible of the available ontologies or vocabularies. In the first step suitable vocabularies or ontologies have to be looked for in order to reuse them. If there is no suitable vocabulary term for a specific purpose, then it should be created from scratch. Metadata publishing usually relies on existing ontologies developed for different purposes and produces a description of the actual content in a machine-usable format.

Data model. In order to create a well-designed Linked Open Dataset, its data model has to be specified, which is composed of the following main tasks. First the available and for the specific task necessary vocabularies and ontologies have to be chosen, they will be specified under the used namespaces. Then the used classes and their relationship have to be specified and finally the possible properties of the entities for each class have to be collected. If it is possible, the property should be a member of an existing vocabulary or ontology. If a necessary property does not exist yet, then it should be created for the given data model. For each property, its meaning, its URI, its domain and range have to be specified. Furthermore in the description of the property its mandatory or recommended role can be marked as well.

In case of BitTorrent Metadata schema the data model uses vocabularies, like Dublin Core, FOAF, schema.org and DBpedia. In the BitTorrent ontology there are three different types of classes:

1. A core class representing the files containing BitTorrent contents.

2. Core classes representing the BitTorrent contents. The number and the type of the used classes for desribing the content depend on the possible types of the available contents in BitTorrent. There is a possible decision here: either to use one class for all types of content or to create separate classes for the different content types (like music, movie, picture, sound or document). In the latter case we suggest to use a parent class that contains details for all kind of contents and subclasses (like movie, music, book) for content type specific details.

3. Contextual classes containing associated information for the core class entities. For example if content metadata contains details about an entity that is distinct from the content (for instance, the date of birth for an author) then these entities should be kept separate from the data about the content

itself. If we think about a scientific publication metadata registry, besides the publications as an ontology class, it's worth to have a separate one for the authors. It can be similar for Places, Timespans and Concepts.

The following figure illustrates the classes, their relationships and some of their properties in the suggested BitTorrent ontology:

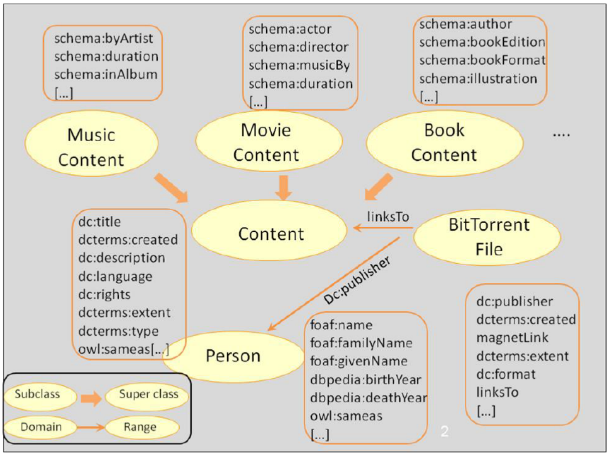

Figure 2: Ontology for search in BitTorrent

In BitTorrent Metadata Elements specification Magnet links can be represented in two different ways. (1) Either they would be objects in the RDF triples in the form of a string literal. In this case a special property should be created for them in the data model. Storing Magnet links in their original form gives a cheap and portable solution. (2) Or a specific class can be created for Magnet link entities. In this case the Magnet link URI scheme can be easily transported to this ontology Class. This provides more functionality, but is more complex regarding handling and storing.

It also needs to be specified how links can be established between the BitTorrent LOD metadata store with other LOD sources. A feasible solution is the use of owl:sameAs property. In this way a correspondence can be set between two entities from different metadata resources. Setting up this correspondence can be the task of the content uploader or it can be supported by an automatized process. If the content uploader determines the RDF links to entities of other data sources, then BitTorrent Disvovery Site should support it by providing possibilities to record the correspondence in the upload form or in the BitTorrent Metadata validation process used for metadata uploading. In case of an automatized process RDF link discovery tools can be applied like Silk – Link Discovery Framework [15], which is a tool designed to find relationships between entities within different data sources and to set explicit RDF links between these entities. In order to realize automatic link generation the SPARQL endpoints access parameters of the feasible data sources have to be specified. Besides it a restriction has to be drawn from the sets of examined resources to smaller datasets, it determines which instances can be interlinked with each other according some specific conditions. These conditions must be fulfilled in order to interlink two entities. The type of link that connects the source and target entities (e.g. owl:sameas, rdfs:seeAlso, foaf:based_near or foaf:page) have to be specified as well.

5. Summary

In the publication we described a new framework for the sharing of open data that provides possibility for sophisticated metadata search on the content. We chose to base the content distribution on existing P2P technologies, namely on BitTorrent protocol and noticed a need for a well-designed search phase. In the paper we studied the various search schemes in the current P2P landscape, proposed a search scheme which is based on Linked Open Data, worked out the data model for BitTorrent Metadata schema and proposed an ontology schema for this data model.

References

[1] JOSEPH, S. P2P metadata search layers. In: Agents and Peer-to-Peer Computing. Springer Berlin Heidelberg, 2005. p. 101-112.

[2] ANDROUTSELLIS-THEOTOKIS, S., SPINELLIS, D. A survey of peer-to-peer content distribution technologies. ACM Computing Surveys (CSUR), 2004, 36.4: 335-371.

[3] RIPEANU, M. (2001, August). Peer-to-peer architecture case study: Gnutella network. In Peer-to-Peer Computing, 2001. Proceedings. First International Conference on (pp. 99-100). IEEE.

[4] MADHUKAR, A., WILLIAMSON, C. (2006, September). A longitudinal study of P2P traffic classification. In Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on (pp. 179-188). IEEE.

[5] NEJDL, W. et al. EDUTELLA: a P2P networking infrastructure based on RDF. In: Proceedings of the 11th international conference on World Wide Web. ACM, 2002. p. 604-615.

[6] COHEN, B. The BitTorrent protocol specification. 2008.

[7] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M., BALAKRISHNAN, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of SIGCOMM 2001.

[8] ZHANG, C., DHUNGEL, P., WU, D., ROSS, K. W. (2011). Unraveling the bittorrent ecosystem. Parallel and Distributed Systems, IEEE Transactions on, 22(7), 1164-1177.

[9] HALES, D., PATARIN, S. (2005). How to cheat bittorrent and why nobody does. Technical Report UBLCS 2005-12, Computer Science, University of Bologna.

[10] WOLCHOK, S., HALDERMAN, J. A. (2010). Crawling BitTorrent DHTs for fun and profit. Proc. of WOOT (Washington, DC, USA, 2010).

[11] KRONFOL, A., Z.: FASD: A fault-tolerant, Adaptive Scalable, Distributed Search Engine. PhD thesis, Princeton University (2002).

[12] JOUNG, Y. J., YANG, L. W., FANG, C. T. (2007). Keyword search in dht-based peer-to-peer networks. Selected Areas in Communications, IEEE Journal on, 25(1), 46-61.

[13] BERNERS-LEE, T.: Linked data-design issues (2006).

[14] Dublin Core Metadata Initiative: `http://dublincore.org/documents/dcmi-terms/` (download 22.04.2014)

[15] VOLZ, J., BIZER, C., GAEDKE, M., KOBILAROV, G. (2009, April). Silk-A Link Discovery Framework for the Web of Data. In LDOW.