P2P MultiRouter

A Report submitted for Distributed systems CS303

By

Rajesh Kumar
Bachelor of Technology, V Semester
Roll No. 15010105

Department of Computer Science and Engineering
Indian Institute of Information Technology Manipur
November, 2017

Abstract

Peer-to-peer distributed hash tables (P2P DHTs) are individually built by their designers with specific performance goals in mind. However, no individual DHT can satisfy an application that requires a "best of all worlds" performance, viz., adaptive behavior at run-time. We propose the MultiRouter, a light-weight solution that provides adaptivity to the application using a DHT-independent approach. By merely making run-time choices to select from among multiple DHT protocols using simple cost functions, we show the MultiRouter is able to provide a best-of-all-DHTs run-time performance with respect to object access times and churn-resistance. In addition, the MultiRouter is not limited to any particular set of DHT implementations since the interaction occurs in a black box manner, i.e., through well-defined interfaces. We present microbenchmark and trace-driven experiments to show that if one fixes bandwidth at each node, the MultiRouter outperforms the component DHTs

Contents

Abstract **ii**

1 Introduction **2**
1.1 What is Peer to Peer (P2P) systems ? 2
1.1.1 Characteristics of a Peer Network 2

2 Background Analysis **3**
2.1 Peer-to-Peer and P2P Networks 3
2.2 Some of the issues facing designers of P2P routing algorithms are: 3

3 Categories : **5**
3.1 Gnutella Routing 5
3.2 Distributed Hash Tables 6
3.3 Semantic Routing: 7
3.4 Freenet 8
3.5 Routing algorithm: 9

4 Advantages **11**

5 Disadvantages **12**

6 Applications : **13**
6.1 Peer-to-Peer Systems 13
6.2 From Paper : 14
6.2.1 Conclusion : 14

Bibliography **14**

Chapter 1

Introduction

1.1 What is Peer to Peer (P2P) systems ?

Peer-to-peer networking is an approach to computer networking in which all computers share equivalent responsibility for processing data. Peer-to-peer networking (also known simply as peer networking) differs from client-server networking, where certain devices have responsibility for providing or "serving" data and other devices consume or otherwise act as "clients" of those servers.

1.1.1 Characteristics of a Peer Network

- Peer-to-peer networking is common on small local area networks (LANs), particularly home networks.
- Both wired and wireless home networks can be configured as peer-to-peer environments.
- Computers in a peer-to-peer network run the same networking protocols and software. Peer networks devices are often situated physically near one another, typically in homes, small businesses and schools. Some peer networks, however, utilize the internet and are geographically dispersed worldwide.
- Home networks that use broadband routers are hybrid peer-to-peer and client-server environments. The router provides centralized internet connection sharing, but files, printer, and other resource sharing are managed directly between the local computers involved.

Chapter 2

Background Analysis

2.1 Peer-to-Peer and P2P Networks

The topology of an overlay network may change all the time. Once a route is established, there is no guarantee of the length of time that it will be valid.

- [1]Peer to Peer (P2P) systems can take many forms. Email, Internet Relay Chat and Napster are all examples of P2P systems.
- Routing on these networks is either centralised or statically configured and is therefore unproblematic.
- Another class of P2P networks is the overlay network.
- Overlay networks build a virtual topology on top of the physical links of the network. Nodes leave and join this network dynamically and the average uptime of individual nodes is relatively low.
- The topology of an overlay network may change all the time. Once a route is established, there is no guarantee of the length of time that it will be valid.

2.2 Some of the issues facing designers of P2P routing algorithms are:

Routing in these networks is therefore very problematic and will be the focus of our report. Some of the issues facing designers of P2P routing algorithms are:

- Scalability
- Complexity

[1]http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group6/ Date 02 Nov, 2017

- Anonymity
- Scalability is a measure of how a system performs when the number of nodes and/or number of messages on the network grows.
- Complexity is the order of steps required for a packet to travel from one host to another in a worst case scenario.
- Anonymity is not a requirement of most P2P networks, however if a network is to be designed to provide anonymity then this is a problem that must be solved at the routing level. We will look at a few examples of routing algorithms from each of these perspectives.

Chapter 3

Categories :

- Gnutella Routing
- Distributed Hash Tables
- Semantic Routing
- Freenet

3.1 Gnutella Routing

Gnutella was arguably the first mainstream overlay network to enjoy widespread use.

1. **Gnutella :** was arguably the first mainstream overlay network to enjoy widespread use.
2. The concept behind it was simple. To join the network a client had to know the address of at least one node already on the network.
3. Once the client had a connection to this node it could then broadcast a ping to find the addresses of other nodes.
4. The basic idea is that each node maintains a connection to a number of other nodes, normally about five.
5. To search the network for a resource the client sends a "Query" message to each of the nodes it's connected to. They then forward the message and when a resource is found the result i.e. resource name and address is propogated back along the path.
6. The number of nodes that get queried can be somewhat controlled using a Time-To-Live counter. This type of routing is the simplest kind possible for an overlay network, it is not however without its problems.

7. Gnutella style routing, or flooding, works well for small to medium sized networks.

8. It has been shown that the cost of searching on a Gnutella style network increases superlinearly as the number of nodes increases.

9. When node saturation occurs the network can become fragmented. Gnutella, therefore, while a simple solution does not scale well. Searching in a Gnutella network is roughly of exponential complexity, as each search will take about n to the power d steps where n is the time to live and d is the number of peers per node.

10. Flooding is obviously not the most optimal solution for routing, and it wasn't long before other P2P routing algorithms emerged which were more efficient.

11. We will discuss a number of these including semantic routing and distributed hash tables. Gnutella wasn't designed to be anonymous and discovering who is making specific resources available is a simple matter of performing a search.

12. One network which was designed specifically to be anonymous is Freenet. Freenet's routing algorithm was designed completely with anonymity in mind.

13. We will take a look at the issues that anonymity presents in terms of scalability and complexity.

14. Chords main advantage is the guarantee that you will get a reply within log(n) time.

15. Also a significant advantage is the lack of redundant overhead.

16. These both give it a huge edge on any flooding algorithm. But in general, given that DHT algorithms store their data references in an organized way, they will always beat flooding algorithms in this area.

17. Chord scales well, given that the search is of order log(n), and also has relatively low complexity because of this.

3.2 Distributed Hash Tables

- Distributed hash table (DHT) algorithms are useful for sharing files or other data across a peer-to-peer network.

- A hash function takes a variable length string of bytes and returns a number that it generates from this.

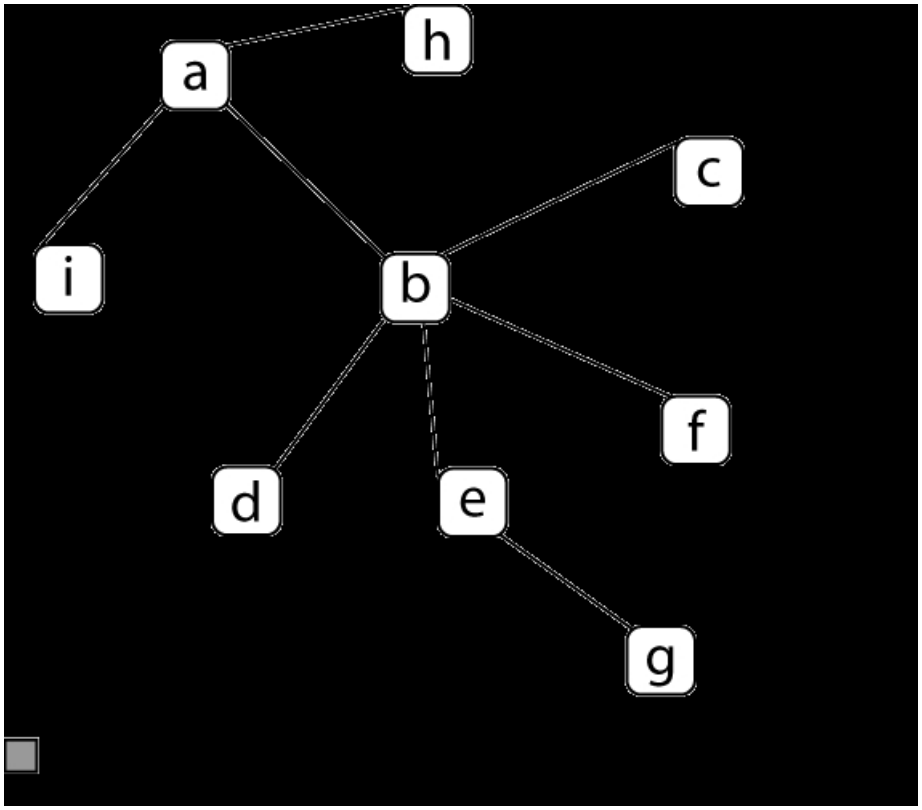

Figure 3.1: Gnutella Routing

- DHT algorithms work by hashing all file/data identifiers and storing their locations in a giant hash table, which is distributed across the participating nodes.
- One research group in MIT has developed a system called Chord, which is a good example of a DHT algorithm.

  Node 1 finger table X Successor(X) 3 4 5 6

  Node 6 finger table X Successor(X) 0 0 2 4

3.3 Semantic Routing:

- Semantic Routing is a method of routing which is more focused on the nature of the query to be routed than the network topology. Essentially semantic routing improves on traditional routing by prioritising nodes which have been previously good at providing information about the types of content referred to by the query.
- In order to be able to search for information on a p2p network semantically the data needs to have a semantic description associated with it, one popular solution is the use of RDF meta-data[1] for this purpose. Tagging

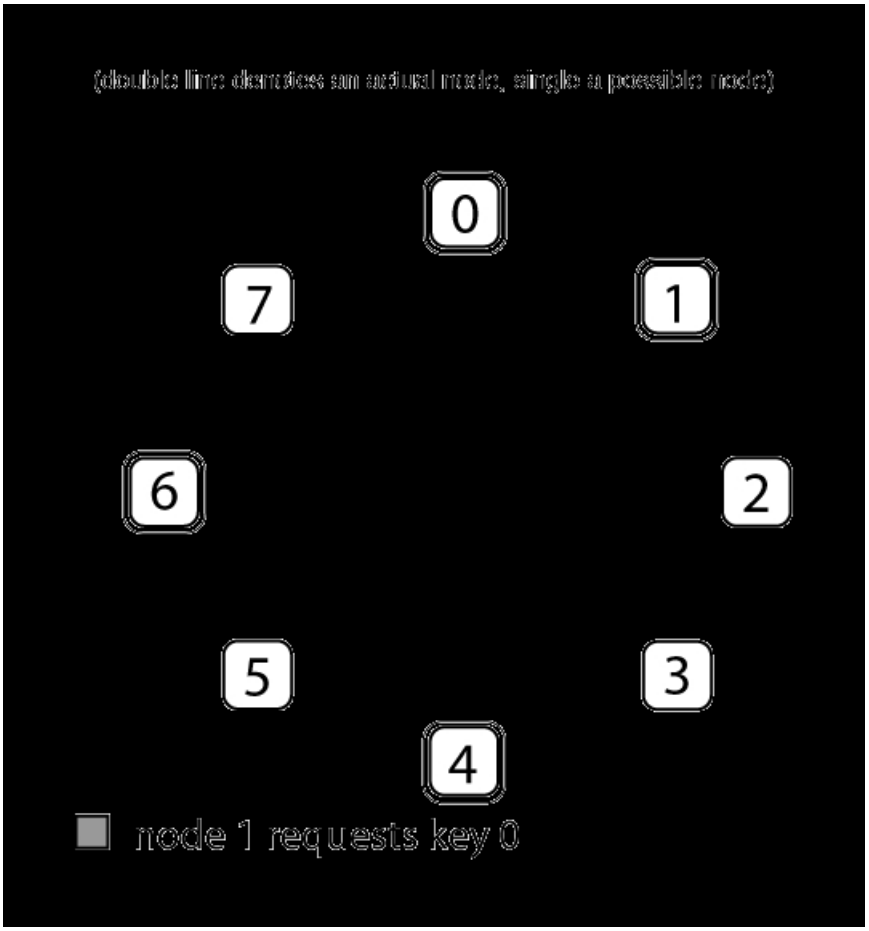

Figure 3.2: Distributed Hash Table

documents/data with RDF would provide a rich 'semantic web' which could be structured in a p2p fashion. A schema-based p2p network such as this would benefit greatly from semantic routing. Semantic routing differs fundamentally from other routing techniques because prospective nodes are selected because of another node's confidence in their ability to respond correctly to a given query irrespective of their position within the network.

3.4 Freenet

- Freenet is a peer-to-peer networking system with very specific goals - both political (anonymity, and freedom of speech) and technical (decentralisation of all network functions and efficient dynamic storage and routing of information).

- Each node on the network acts as a data store, and has no say in what data it contains - any other node may read from or write to this. This

enables the network to act as a distributed file-system. The data on each node is encrypted, providing further anonymity.

- Files on the system are referenced and located using hash-keys. In order to search for a file a user must know it's hash-key, or at least how to calculate it. Once this is known the user sends a request to their own freenet node. Each request has a hops-to-live value, and a randomly generated id so it can be recognised and rejected by nodes which have seen it before. When a node receives a request it first checks it's own data-store to see if it contains the relevant file. If it does then it returns the file along with a message identifying it as the source of the data. If not it decrements the hops-to-live value and forwards the request to one of the neighbours in it's routing table. If this request fails then it asks another neighbour from the routing table. If none of it's neighbours return a positive response then it returns a failed message itself. If the hops to live value is exceeded then a failed message is also returned. If a positive response is received from a node then this node sends the data to the node it received the request from, and caches a copy of the data in it's own data-store. If the data-store is already full, then the least recently used files are deleted to make room for the new one. In this way, more popular data will be cached on multiple nodes throughout the network, and data that no-one requests will eventually be removed entirely.

3.5 Routing algorithm:

- Choosing the next neighbour to request content from is an important part of freenet's behaviour, as it is this that it allows it to adapt to usage patterns and network changes.

- Each node has a routing table which contains the address of it's neighbours, and also their performance in returning particular keys. This performance rating was initially just an indication of how successful the node had been in retrieving the key, but has been improved to include response-time and transfer time. When a node receives a request for a particular key it searches it's table to find the node which was most successful in returning a similar key. It is important to note that it is the keys which are being compared for closeness, not the files, so there is no semantic similarity implied. It then requests the file from this node.

- When a node receives a successful response from another node, it will create a new entry in it's table associating the source node with the requested key. An emergent property of this is that nodes will start to specialise in particular keys. When the network is set up there will be no performance records for any nodes, so requests are sent to node which

is essentially chosen at random. If it successfully returns the requested file, then an entry will be created for that node with the relevant key. Requests for similar keys will then be routed towards the successful node, and as a result it will eventually start to specialise in these keys.

Chapter 4

Advantages

- Popular data is replicated throughout the system.
- Data is distributed anonymously and freely, the main goals of the system.
- The routing algorithm is designed to adapt and improve efficiency over time.

Chapter 5

Disadvantages

- People are hesitant to donate part of their hard-drive to the system, particularly when it could be used to store information they don't approve of .
- The network is not easily searchable - users need to know the key to find a file.
- The network is slow, as the data must pass through all intermediate nodes. Searches can be particularly slow because they don't use multicasting.

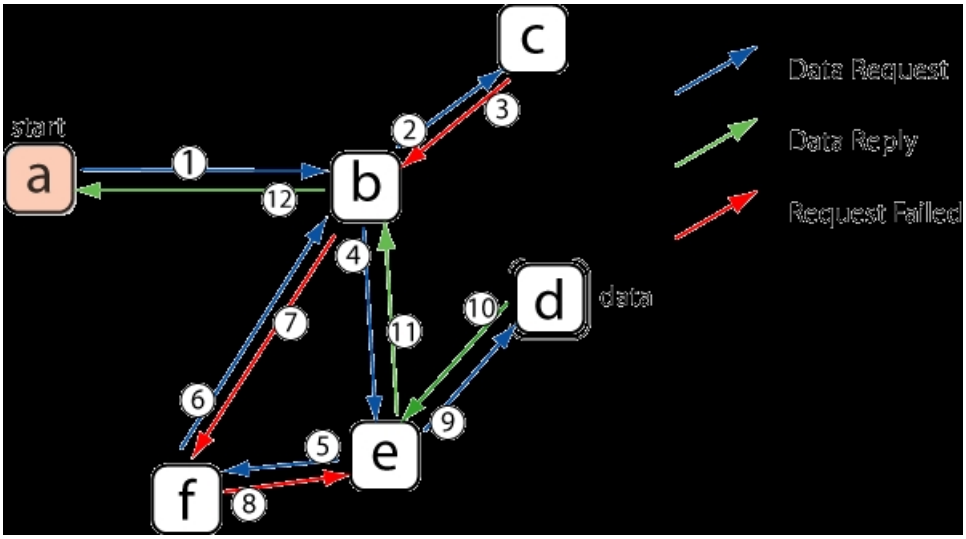

Figure 5.1: Freenet

Chapter 6

Applications :

6.1 Peer-to-Peer Systems

- [1]Cornell faculty have done extensive work in the Peer-to-Peer networking area, ranging from file sharing to media streaming to network monitoring. Emin Gun Sirer and his students have created a large number of P2P systems, including Blindfold (IPTPS 2010), a scheme to ensure that the operators of content aggregators are completely blind to the content that they are storing and serving, thereby eliminating the possibility to censor content at the servers ;

- Antfarm (NSDI 2009), a content distribution system based on managed swarms;

- Octant (NSDI 2007), a system for geolocation of Internet hosts; and

- Beehive, a peer-to-peer replication system on which a new DNS, an Internet-scale Publish-Subscribe system and a new content distribution network were built (NSDI 2006, SIGCOMM 2004). Sirer's Credence system (NSDI '05) for determining authentic content on peer-to-peer systems was deployed on the Gnutella network and led to a large-scale user study.

- The Karma system explored peer-to-peer currencies long before Bitcoin. Van Renesse developed Fireflies, a Byzantine-tolerant P2P overlay network (Eurosys 2006). Van Renesse and Birman developed the highly scalable Astrolabe network monitoring system (IPTPS 2002, ACM TOCS 2003), now used at a major e-retailer. Birman's Kelips system was the first one-hop DHT, and his Bimodal Multicast protocol was unusual in using P2P communication as a tool in a reliable multicast protocol. Weatherspoon designed and implemented the Antiquity system, a secure P2P storage facility (Eurosys 2007).

[1]https://www.cs.cornell.edu/research/systems Date 02 Nov, 2017

6.2 From Paper :

6.2.1 Conclusion :

In this paper, they described Blindfold, a system that enables users to upload to and search a public key-value store without revealing the true keys or values to the store or third parties. The system works by partitioning and chaining upload and search operations into a series of key-value operations across servers in different administrative domains. The connection between the servers is obscured and protected by captchas. We showed that the system is simple and feasible with a prototype implementation, and we have found from experience with the system that it is surprisingly unintrusive to the user and easy to use.

Bibliography

[1] R. S. Peterson, B. Wong, and E. G. Sirer, "Blindfold: a system to" see no evil" in content discovery." in *IPTPS*, 2010, p. 1.