Analysis of Overlay Network Impact on Dependability

Piotr Karwaczyński, Jan Kwiatkowski
Division of Computer Science
Wroclaw University of Technology
50-370 Wroclaw, Wybrzeze Wyspianskiego 27, Poland
{piotr.karwaczynski, jan.kwiatkowski}@pwr.wroc.pl

Abstract

Recently, peer-to-peer systems have become widely accepted and are probably the most recognizable examples of distributed applications. As they are maturing and their functionalities are becoming increasingly complex, the need for dependable solutions arises. In particular peer-to-peer systems' dependability is immensely influenced by their virtual overlay networks. The paper presents the results of analysis of diverse overlay networks with respect to their support for dependability.

1. Introduction

Nowadays distributed processing using computers connected via computer networks is widely accepted both for high performance scientific computing and for general-purpose applications. It is becoming one of the most attractive and the cheapest ways to share resources and to increase the computing power. However, taking advantage of this technique requires specialised software. Many attempts have been made to-date to address this issue. Among them are peer-to-peer (p2p) systems, whose foundations are diverse overlay networks.

A peer-to-peer overlay network is a logical layer built on top of computer network infrastructure. We can distinguish two fundamental components of overlay networks: their topologies and algorithms used to discover resources in scope of the specific P2P system. Pairs of topology and resource discovery algorithm constitute overlay networks having different properties related to dependability. In general, we can distinguish five different types: centralized topology with centralized directory model, decentralized topology with two different resource discovery algorithms: search query flooding or search based on resource routing and two hybrid topologies that utilize both previously mentioned discovery algorithms. This classification of overlay networks will be used in the paper. It has been chosen since it (1) divides p2p overlay networks into groups, where each group encompasses systems having similar dependability-related properties and (2) spans the space of all contemporary p2p overlay networks.

In general, dependability can be defined as the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers [3]. To meet our needs, we refined the above definition by presenting a set of dependability attributes related to peer-to-peer overlay networks domain.

The goal of the research is to analyse an impact of different types of overlay networks on a peer-to-peer systems' dependability. The results of the analysis will be utilized in a DeDiSys project ("Dependable Distributed Systems"), funded under 6 Framework Programme of European Community. One of the project's main objectives, related strongly to dependability of overlay networks, is to introduce an availability–consistency tradeoff into distributed systems. So, we are trying to find the most suitable type of overlay network (the most dependable at present and easy to extend) for usage in the DeDiSys project.

In this paper we will address the impact of overlay networks on dependability. It is organised as follows. Section 2 describes different concepts of the peer-to-peer systems' dependability. Section 3 tells more about the overlay networks presenting different approaches to peer-to-peer systems and their relations to dependability. Section 4 presents the results of analysis on dependability. Finally section 5 outlines the work and discusses the ongoing works.

2. Dependability in peer-to-peer systems

A system's dependability expresses the expectations (trust) of its users regarding how the system meets their functional and non-functional (e.g. quality-of-service) requirements. It is worth emphasizing the role of the users – their views are crucial in deciding which system's properties are in fact valuable. It may be stated that the regular users of up-to-date large-scale peer-to-peer system expect to:

- connect to the p2p network immediately and anonymously,
- share the capabilities of their nodes without complex or time-consuming configuration phase,
- effectively look for data and services interesting to them,
- effectively make use of the data and services according to their specific needs.

Such conclusions may be drawn easily after even cursory examination of the most popular peer-to-peer systems as opposed to the systems rejected by the Internet community.

As functional requirements are specific for given systems, we are considering dependability only in terms of non-functional requirements. Typically, dependability is described as a set of non-functional system properties, also known as dependability attributes. There do exist a number of proposed alternatives [1], [2]. They establish a strong, widely accepted comprehensive baseline. However, the properties must be selected and defined according to a specific system's context and its users' needs. Hence we propose the alternative of dependability attributes adequate for peer-to-peer domain: reliability, availability, scalability, integrity and adaptability. They are defined below.

Reliability: continuity of correct service. Lack of failures (i.e. inconsistencies with a service specification or deviations from Service Level Agreement) is a foundation for reliability. Moreover, a system should be able to operate as a whole despite minor attacks or failures influencing its topology and connections.

For peer-to-peer overlay networks it means there are always (if required) nodes available and able to communicate and cooperate. There are some typical areas, where a system's reliability may be weakened, i.e. keeping and updating addresses of network participants, relying on single points of weakness.

Availability: readiness for usage. As peer-to-peer overlay networks obtain, propagate and deliver queries and responses, they are available if every query is answered and every response is adequately delivered in a reasonable (satisfying for a user) time.

A very important aspect of peer-to-peer systems is their overlay network resource discovery model. It strongly influences the users' satisfaction level and eagerness to use system services, as it is responsible for how long the resource searching is performed and whether the search results are accurate.

Scalability: the ability to operate without a noticeable decline in performance despite the changes in a number of nodes constituting the system.

The topology as well as resource discovery algorithm of overlay networks has an immense influence on the scalability of a peer-to-peer system. As peer-to-peer systems are typically pervasive, composed of even hundreds of thousands of nodes and more, scalability property must not be ignored. It is mainly imposed by such system aspects as: hierarchical or flat topology, communication overhead of resource discovery algorithm, reaching resources localized behind firewalls/NATs, awareness of resource locality.

Integrity: the ability to keep a system state and information on it coherent and up-to-date among all the entities constituting the system.

Peer-to-peer overlay network state is usually very complex due to the large number of nodes. There are two viewpoints on the state of such a network: global – rarely known to any single node, and local – covering only a small part of the network. As global view is usually impossible to obtain, local views are used. However, a set of local views is only a rough estimation of the system state, strictly dependant on their staleness. The more up-to-date local information on peers constituting a system and their resources, the better the estimation of a global system state.

Adaptability: the ability to propagate information and services effectively even in case of remarkable changes of the working environment.

As peer-to-peer systems are usually highly dynamic, the joining and leaving of nodes is commonplace. In the main a system is highly adaptable if its components exchange the up-to-date information on its structure in a very dynamic manner.

Among dependability attributes there are many others, not listed above, like maintainability (the ability to undergo modifications of a system software), safety (the ability to operate without catastrophic consequences on users and their environment), etc. However, they are not specifically connected with overlay networks, thus are out of the scope of our interest.

3. Peer-to-peer overlay networks

3.1. Characteristics related to dependability

A peer-to-peer overlay network is a virtual network overlay built on top of the existing physical network infrastructure (figure 1).

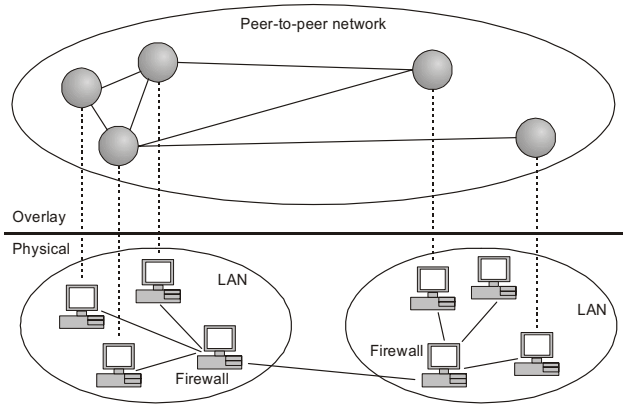

Figure 1. Mapping between physical and virtual networks

It logically connects all peers, directly or indirectly, in a peer-to-peer network. Such an overlay has its own topology, independent from the physical network, its own routing and resource discovery algorithms. Peers use it as a logical communication layer. It is introduced in order to

support specific distributed or even decentralized algorithms and provide adequate abstractions. Moreover, it allows them to overcome common communication problems such as those encountered during cooperation over networks built on different network protocols and having diverse addressing schemes.

We may distinguish two main characteristics allowing the classification of peer-to-peer overlay networks: topology and resource discovery algorithm. Each of them has a strong influence on the dependability of a peer-to-peer system as a whole. Whereas they are significantly interdependent, different configurations are possible.

Topology. There are a few common topologies present in peer-to-peer systems (figure 2).

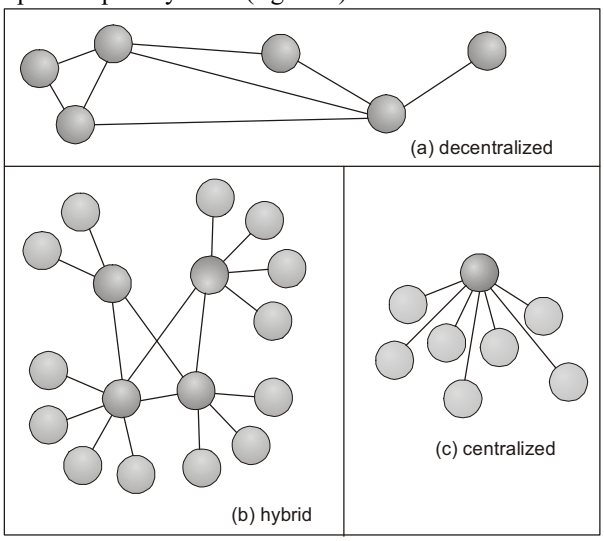

Figure 2. Peer-to-peer topologies

A centralized topology is strongly based on a client-server paradigm. However, it is worth emphasizing that there may exist some variations, allowing peers acting as clients to be autonomous to different degrees: a server may serve only as a database of indices to resources, a bootstrapping node or, on the contrary, it may exclusively control all the system. Common examples are SETI@home and Napster.

In a decentralized topology all peers are equal and autonomous. There is no central point responsible for system maintenance or performing any particular task exclusively. Systems adopting this type of topology may have a twofold nature: they can be structured, where connections between peers are by some means organized (e.g. DHT-based: Chord, Kademlia) or unstructured ('small-world' structures: Gnutella, Freenet).

A hybrid topology is a mixture of centralized and decentralized topologies. Typically, a set of distinctive peers (super-peers), having better hardware performance or network bandwidth, forms a distributed subsystem, whereas all other peers treat super-peers as servers, creating centralized, satellite subsystems. Examples are JXTA and Kazaa.

Resource discovery algorithm. Algorithms supporting resource discovery in peer-to-peer networks generally are consistent with one of three models [4]: centralized directory model, flooded requests model or resource routing model.

A centralized directory model is rather straightforward. There exists a well-known node, serving as a central point, where all the information on resources available in a system is stored. A peer, looking for some particular resource, sends a query request to the central peer and receives information on possible sources, if any. Then, the communication between the source peer and requesting peer may be carried out directly.

In a flooded requests model peers have no information on resources available. They are made to probe the network, usually using broadcast techniques. Such an approach exploits the network resources to their limits and there are a few methods to improve it. The most popular is using time-to-live query request parameter expressing a maximum number of query hops, hence possibly reducing the network traffic. Another one is two-layered system architecture, related to hybrid topology, where flooding is carried out only among super-peers being more available and powerful than usual peers, on behalf of a peer sending a query request to its super-peer. Also, caching may significantly decrease the searching overhead in a network.

The most advanced algorithms are those based on a resource routing model. Among them, a Distributed Hash Table (DHT) approach is the most popular. The general idea is to associate resources with keys using hash function, to store certain ranges of keys distributed among peers and to provide suitable indexing mechanism to store and retrieve resources based on their keys. However, except DHT, there are other approaches, e.g. based on different proximity metrics.

3.2. Overlay network impact of peer-to-peer systems on their dependability

In order to examine the impact of overlay networks on dependability of peer-to-peer systems, we briefly present real-life applications, both widely used or rather educational, in terms of their topologies and resource discovery algorithms. Among presented pairs of topology – resource discovery algorithm, we discuss:

- centralized topology, centralized directory model,
- hybrid topology, flooded requests model,
- hybrid topology, resource routing model,
- decentralized topology, flooded requests model,
- decentralized topology, resource routing model.

On these grounds, conclusions on their dependability level will be made.

3.2.1. Centralized topology, centralized directory model. Typical examples of this kind of peer-to-peer

systems are SETI@home [5] and Napster [14]. SETI@home, a distributed computing system, is focused on analyzing large number of narrow-bandwidth radio signals from space. Though it is based on a single server, we categorize it as a peer-to-peer system due to high autonomy of its clients. The server divides data from a radio telescope into fixed-size work units, distributes them via Internet to millions of registered client programs and collects the results of their analyses. The clients make all computations required to analyze data during their idle periods. Napster, a file sharing system, supports searching of MP3 files among its clients. The role of the central server is to index music files of registered users and answer the clients' browse and search queries (figure 3). The files are downloaded directly between clients.

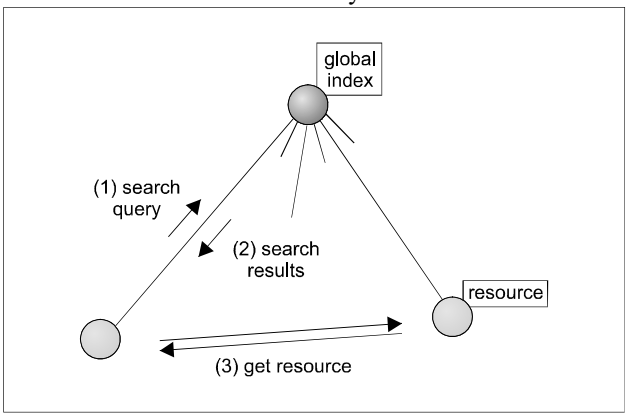

Figure 3. Resource discovery in Napster

The centralized topology imposes the existence of a server. Its responsibilities may differ. In SETI@home, a server distributes data and collects the results of analyses. Napster's server indexes files and performs searching at the request of its clients. In both cases, a server is a single point of failure. When the server becomes out of order, the system crashes.

The presence of a single point of failure makes this type of peer-to-peer systems unreliable. It was proved in the Napster's case, where the existence of the system turned out to be completely dependant on centralized decisions. For the same reason, centralized systems are very vulnerable to planned attacks. Similarly, this topology supports adaptability only within limits. A dynamics of joins and leaves of peers in a system is directly limited by server's performance and bandwidth. Taking scalability into account, it is the least scalable topology of peer-to-peer systems, even if in some applications it may turn out to be sufficient.

Despite the fact that this approach seems to have an obvious bottleneck, SETI@home and Napster proved it is able to serve hundreds of thousands of users in a satisfactory way. However, centralized topology is unsuitable if large pieces of data might be exchanged between peers and a server or complex communication processes could take place.

The significant advantage of a centralized topology in terms of dependability is its support for integrity. It provides high level of integrity as all the information related to the state of the system is concentrated in one place. Hence, keeping up-to-date information on participating peers and their resources is easy to maintain.

The algorithm of a resource discovery is very simple. A peer asks a centralized directory for the existence and location of a particular resource. As an answer, a peer obtains the information concerning whether the resource is available and who its owner is (the server or other peer).

The simplicity of resource discovery results in high availability provided that the central server is not overloaded. Searching covers all the system's scope without any communication overhead. The search results are delivered with minimal latencies.

3.2.2. Hybrid topology, flooded requests model. The typical representative of a hybrid topology combined with flooded requests model is Kazaa. It is an example of a file sharing application. Nowadays Kazaa is one of the most popular distributed systems deployed in the Internet. As its protocol is proprietary and encrypted, there are only general information [7] and measurement-based presumptions [8] available.

Kazaa has two classes of peers. It uses a decentralized network of super-peers (about 1% of all peers) acting as servers for highly dynamic sets of ordinary peers organized in a centralized manner. Super-peers are selected from ordinary peers in a run-time, as those having better CPU power, bandwidth, up time, non-local IP. They form a decentralized network, where the connections are often shuffled. Similarly, ordinary peers frequently exchange lists of super-peers with each other. When a peer obtains the list, it uses data to improve its local super-peers list in terms of locality.

When a peer connects to the network, it selects a super-peer having a low average workload. Then a peer uploads metadata on files it is sharing including file name, file size, content hash (unique signature of a file), and description. Super-peer does not cache metadata of a peer after it disconnects.

Super-peers might be perceived as server-like distinctive nodes. However, their overlay has a strong support for self-organization as super-peers are selected in a run-time on the basis of their capabilities. Thus reliability is satisfactory.

A two-tiered topology significantly supports peer-to-peer system's adaptability. Additionally, the network is highly adaptable as peers frequently exchange information on their neighbors, and, moreover, are able to optimize their connections in terms of locality and workload. Such dynamics positively influences integrity of local views on system's topology. Frequent exchanges of information allow keeping lists of super-peers up-to-

date. Moreover, indexing only directly available metadata induces that with a high probability the local viewpoints on system's resources are not stale.

A resource discovery algorithm makes use of super-peer decentralized network (figure 4). A peer sends a keyword-based query to its parent super-peer. The super-peer uses its own index of resources as well as propagating the query into a small subset of the decentralized network. When satisfactory results are found, a given super-peer returns IP addresses and resource metadata to the peer. Afterwards, the peer may request a file directly from its source, identifying a given file with its content hash.

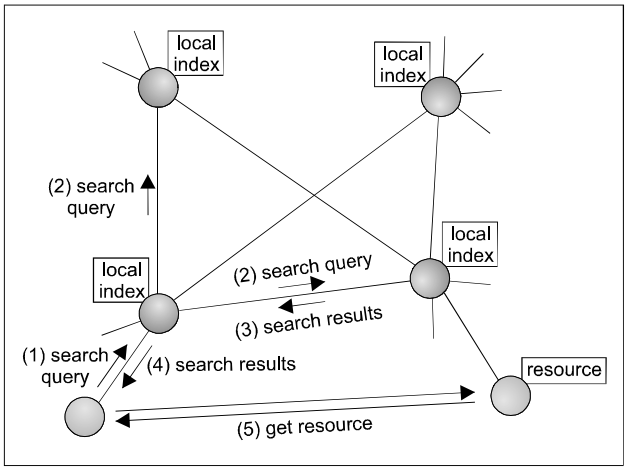

Figure 4. Resource discovery in Kazaa

Searching only in the subsets of the decentralized network results in a serious reduction of the scope of flooding. But on the other hand, the search will not cover all the network resources. To improve it, frequent rewiring of a super-peer network takes place, allowing a larger range of the network to be searched.

The reduction of flooding into the subnetwork of super peers combined with a rewiring of the decentralized network, indexing resources in local super-peers, connecting peers to super-peers based on locality and super-peer average workload keep a response time satisfactorily short. Moreover, Kazaa Lite (a specific Kazaa client), alters the rudimentary searching scheme in order to examine the broader scope of the network. It propagates the query not only to its parent hub, but also to other known super-peers. Summing up, Kazaa has a high level of availability, both in terms of search results and search time.

Kazaa proved to be scalable up to millions of peers. Moreover, there are no visible scalability limits. It was achievable mainly due to its self-organizing, two-tiered architecture, locality awareness and limited scope of network flooding.

3.2.3. Hybrid topology, resource routing model. Another example of a peer-to-peer overlay making use of a hybrid topology is JXTA overlay network [9]. As opposed to Kazaa, it implements a resource routing model. JXTA is an open platform and its overlay infrastructure may be overwritten in order to tailor it to particular needs. However, here we present its default implementation.

A topology of JXTA is two-tiered. A lower tier comprises ordinary peers. A higher tier consists of two independent decentralized networks: rendezvous peers network and relay peers network.

The rendezvous peers network provides a low-level resource discovery mechanism. It comprises rendezvous peers acting as hubs for ordinary peers and maintaining indices of their resources. They maintain an ordered (by peerID) list of known rendezvous neighbors (Rendezvous Peer View, RPV) and periodically exchange random parts of the list with random rendezvous neighbors. In addition, they send a heartbeat message periodically to their nearest neighbors. Every peer may become a rendezvous peer providing it has the right credentials.

The rendezvous peers network is organized as a loosely-coupled network based on a loosely-consistent DHT combined with a rendezvous walker. The walker's role is to support searching and garbage collect out-of-sync indices. This approach is a compromise between structured (DHT) and unstructured (random) topology.

The relay peers network provides the ability to send messages to unreachable (firewall/NAT) or temporarily unavailable peers. It utilizes the same mechanisms as the rendezvous peers network (a loosely-consistent DHT, a rendezvous walker). Relay peers maintain states for peers that cannot be directly reached, for an agreed period of time (a lease).

Similarly to Kazaa, rendezvous or relay peers might be perceived as server-like distinctive nodes. However, their networks are significantly self-organizing. Every peer may become a two-tier peer, despite its capabilities and location. Thus the two-tier networks are satisfactorily reliable.

The peers are able to exchange information on their views on networks' topologies in a highly dynamic manner. Furthermore, a garbage collection mechanism for out-of-sync indices and no preconditions for becoming rendezvous peer improve JXTA adaptability.

The frequent exchanges of information on neighbors in rendezvous and relay peers networks positively influence integrity of local views on a system's topology – with high probability the RPVs comprise the alive rendezvous peers. In addition, an integrity of local views on a system' resources is reasonably high as rendezvous peers maintain metadata indices instead of caching them. Moreover, it is also supported by a limited-range walker acting as a garbage collector of stale indices.

A peer P_a publishes its resource metadata (advertisements) on its parent rendezvous R_a, utilizing SRDI (Shared-Resource Distributed Index) service. SRDI

indexes each advertisement using a predefined number of keys (e.g. name, ID). A rendezvous R_a uses the DHT function to map the index to a rendezvous R_i from its local RPV and forward it there. Moreover, it can replicate the index to the nearest (in terms of RPV) neighbors of R_i: R_{i-1} and R_{i+1}.

A peer P_b, looking for a given resource, sends a query to its parent rendezvous R_b. If R_a and R_b have consistent RPVs, (1) SRDI on R_b computes the value of the DHT function for a given query and, accordingly, R_b forwards the query to R_i (figure 5). Afterwards, R_i forwards it to P_a. Finally, P_a responds directly to P_b.

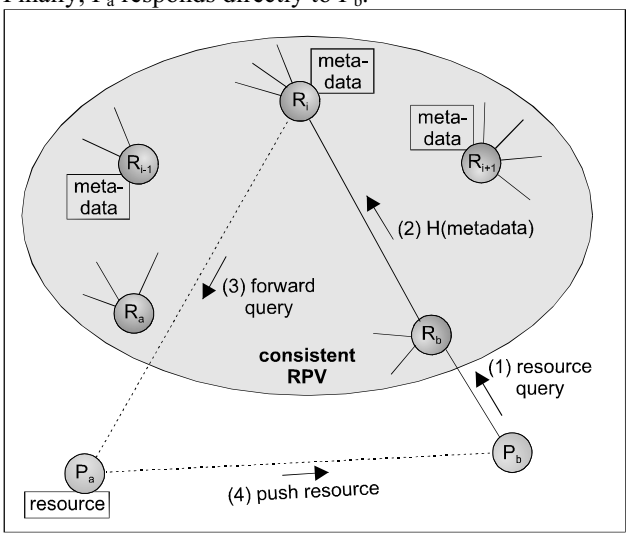

Figure 5. Resource discovery in JXTA (consistent RPVs)

If RPVs on R_a and R_b slightly differ, (2) the DHT function may return a value pointing at a rendezvous R_{i+1} (R_{i-1}) which does own a replica of the given index (figure 6). If inconsistencies in RPVs are significant then the limited-range walker mechanism is utilized to traverse rendezvous from RPV of R_b, starting from R_k (pointed by DHT function) and proceeding in both up and down directions (figure 7). The walker continues traversing till the index is found, the query hop count (time-to-live parameter) is reached or there are no more rendezvous in the given direction. In order to minimize the number of walker's steps, a special hash function is utilized. The function attempts to keep the relative positions of rendezvous in RPV roughly the same despite changes in rendezvous peers network.

System's availability, as opposed to unstructured approaches, is positively influenced by loosely-consistent DHT. On the other hand, the fact that it is only loosely-consistent resource routing and hence it has to be supported by rendezvous walker algorithm may impose noticeable overhead on the general performance.

Nevertheless, such a solution compromises the approaches, where structure maintenance costs are high with those bearing high costs of resource discovery. JXTA does not take locality of peers into account. As a matter of fact, it replicates resource indices based on their proximity, but it is only logical proximity.

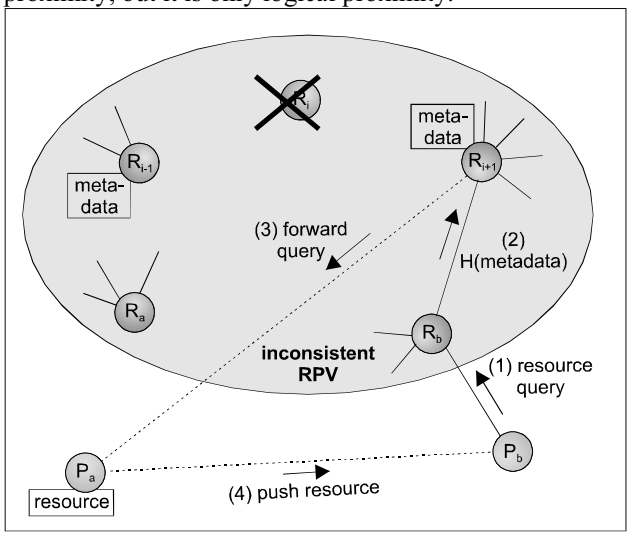

Figure 6. Resource discovery in JXTA (slightly inconsistent RPVs)

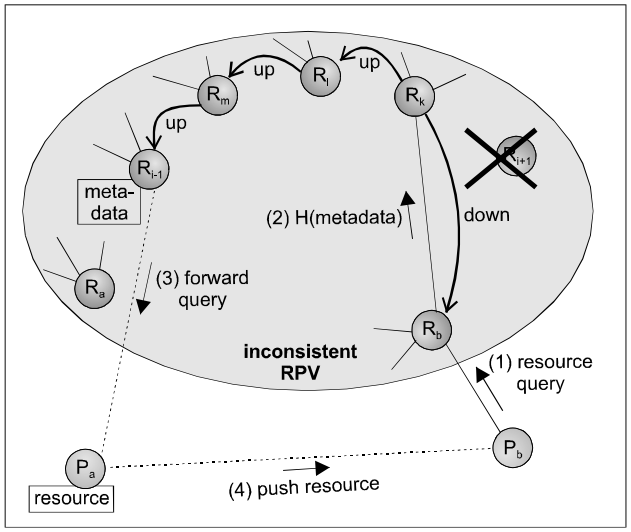

Figure 7. Resource discovery in JXTA (significantly inconsistent RPVs)

Two-tiered topology and loosely-coupled network of relay peers, being proxies for peers behind firewalls/NATs, positively influence scalability. Furthermore, the resource discovery algorithm, that avoids network flooding, does not block system's growth.

3.2.4. Decentralized topology, flooded requests model. Gnutella, a file sharing application, has been one of the first modern peer-to-peer applications built upon decentralized topology. During resource discovery, its network is flooded with search queries. In the primary Gnutella model, each peer can act as a server and a client

at the same time. In this paper, only this model is considered.

Gnutella's topology is flat. There are no inequalities among its peers and there are no single points of failure. Hence the network is extremely reliable. As long as there are two Gnutella peers, the network will continue to exist. Additionally, system's adaptability is unquestionably supported by this type of a topology. Even very dynamic joins and leaves of peers are not harmful as they do not overload any single points in the network.

The search process does not utilize any caches nor indices. The search queries are propagated to direct owners of resources. Hence the search results are as accurate as possible. Thus views of the peers on resources deployed in the network are remarkably exact. On the other hand, there are no specific mechanisms supporting integrity of local views on system's topology. As a result, these views are, to a large extent, inconsistent.

A node that wants to participate in the Gnutella network must connect to any of the existing and active Gnutella peers. Once a peer joins the network, it is able to communicate with adjacent peers and accept any new nodes trying to connect to the network.

A resource discovery is initialized by sending a search query to all adjacent peers (figure 8). Afterwards, it is propagated as long as its time-to-live parameter, being decremented in every step, reaches zero. A peer that received a query, and has a given resource, contacts a searcher directly. Finally, a resource may be downloaded.

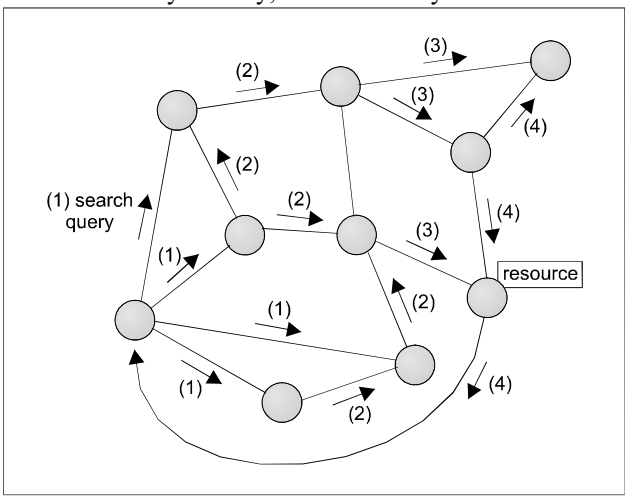

Figure 8. Resource discovery in Gnutella

Every initial search query generates immense network traffic. When the number of peers participating in the network increases, the growing number of search queries negatively influences availability. The scope of the network covered by search is significantly large, but the search time is greatly unsatisfactory. Moreover, flooding the network with search queries reduces Gnutella scalability. Not only do the latencies become noticeable, but also unacceptable for the end-users.

3.2.5. Decentralized topology, resource routing model. As an example of peer-to-peer overlay networks combining decentralized topology with resource discovery model, Chord and Kademlia [11], [13] protocols are demonstrated. These protocols are based on the concept of distributed hash table (DHT) built on top of the overlay. They were introduced as an improvement of a flooded requests model to provide a better efficiency in querying.

The topologies of these networks are pure decentralized without any centralized control or hierarchical organization. Each node runs software with equivalent functionality. Decentralized topology imposes good reliability of these networks as there are no centralized control points. Similarly, it positively influences their scalability.

A basic algorithm provides only two simple primitives: *get(key)* and *put(key, value)*. All the keys constitute a keyspace, and each peer is assigned a small segment of this space. Therefore, a lookup request for a key simply entails finding the peer responsible for the value related to the given key. Using a particular key from DHT, it is able to retrieve the associated value from the proper node. Since the lookup is efficient and straightforward, the level of availability may seem to be high. However, in fact it depends on the consistency of information deployed among the nodes.

Nodes in the Chord network logically create a ring (figure 9). Each node is identified by its m-bit *id* and maintains a so-called finger-table that contains at most log *n* successors from the ring, selected using the following formula: $(id + 2^{k-1}) \bmod 2^m$ for each $k = 1,\ldots,m$. It contains more logically close (in terms of their IDs) then distant nodes. It means that a node can route to the destination in log *n* hops. Each step cuts the distance to the destination by half.

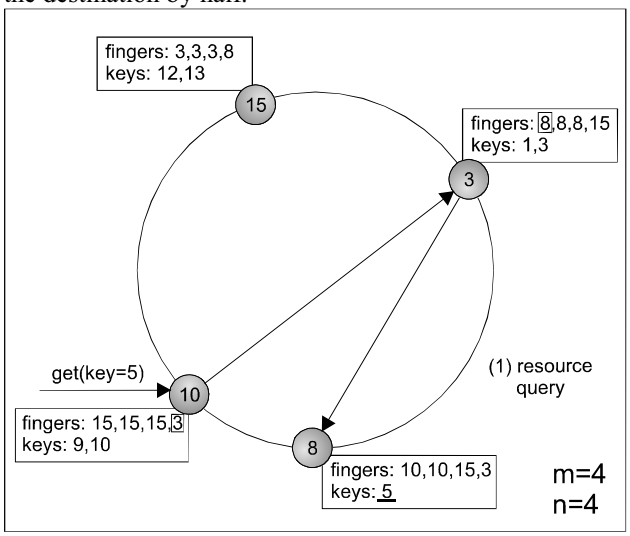

Figure 9. Resource discovery in Chord

During a lookup procedure two cases can appear:

(1) When the node is in the finger table, it can be returned immediately.

(2) If the node is not in the finger table, the closest node to the searched one is asked to find the desired node. By forwarding the lookup, a node nearest to the target is found, and thereby more knowledge is available.

Although the lookup is strongly optimized, the operations of joining and leaving the network require a significant number of control messages ($\log^2 n$). The routing information spread over the network must be kept up-to-date in order to ensure efficient lookup. To address it, a stabilization procedure on every node is invoked periodically. The complexity of handling joins and leaves negatively influences the network's adaptability. Moreover, it entails consistency among local views on system's topology, but with delays.

Another peer-to-peer system which uses DHT tables is Kademlia. In comparison to Chord, the Kademlia protocol minimizes the number of configuration messages as it spreads configuration information automatically as a side effect of key lookups. It improves the network's adaptability and integrity. In addition, Kademlia introduces an alternative eXclusive OR metric for distance between points in the key space. Its further properties are similar to those of other peer-to-peer systems based on DHT tables.

In general, algorithms used by above networks are both structured and symmetric. It means that they may not be vulnerable to individual node failures. Additionally they can implement application level multicast to ensure better scalability and handling network failures.

4. Comparison of overlay networks dependability

The conclusions on dependability of different overlay networks, drawn from the chapter 3 are presented in the tables 1 - 5.

Table 1. The summary of overlay networks reliability

Overlay network	Reliability
centralized topology, centralized directory model	single point of failure (5)
hybrid topology, flooded requests model	self-organizing network of super-peers (3)
hybrid topology, resource routing model	self-organizing network of rendezvous/relay peers (3)
decentralized topology, flooded requests model	all peers are equal (1)
decentralized topology, resource routing model	all peers are equal (1)

Table 2. The summary of overlay networks availability

Overlay network	Availability
centralized topology, centralized directory model	centralized resource directory (1)
hybrid topology, flooded requests model	overhead of flooding (5)
hybrid topology, resource routing model	well-balanced compromise based on loosely-consistent DHT (3)
decentralized topology, flooded requests model	overhead of flooding (5)
decentralized topology, resource routing model	joins and leaves invoke wide-area inconsistencies (3)

Table 3. The summary of overlay networks scalability

Overlay network	Scalability
centralized topology, centralized directory model	limited server's performance and bandwidth (5)
hybrid topology, flooded requests model	overhead of flooding (higher tier) (3)
hybrid topology, resource routing model	two-tiered topology and loosely-consistent DHT (1)
decentralized topology, flooded requests model	overhead of flooding (entire network) (5)
decentralized topology, resource routing model	efficient search algorithm (1)

Table 4. The summary of overlay networks integrity

Overlay network	Integrity
centralized topology, centralized directory model	*resources*: global, centralized index, may be stale (5) *topology*: simple, with one centralized hub (1)
hybrid topology, flooded requests model	*resources*: regional indices (3) *topology*: two-tiered, dynamic, with up-to-date information on super-peer nodes (3)
hybrid topology, resource routing model	*resources*: regional indices and garbage collection mechanism (3) *topology*: two-tiered, dynamic, with up-to-date information on super-peer nodes (3)
decentralized topology, flooded requests model	*resources*: neither indices nor caches – no misleading search results (1) *topology*: unstable, random, dynamic (5)
decentralized topology, resource routing model	*resources*: regional indices (3) *topology*: dynamic, delays in reaching stable states (3)

Table 5. The summary of overlay networks adaptability

Overlay network	Adaptability
centralized topology, centralized directory model	limited server's performance and bandwidth (5)
hybrid topology, flooded requests model	exchanges of information on super-peers; optimization of connections (3)
hybrid topology, resource routing model	exchanges of information on; rendezvous/relay peers (3)
decentralized topology, flooded requests model	completely ad-hoc networks (1)
decentralized topology, resource routing model	overhead related to joins and leaves (3)

The columns represent dependability attributes and the rows – selected types of overlay networks in terms of their topology and resource routing algorithms. In the column that represents integrity, two different aspects are addressed: the consistency of views on resources with a real resources present in the system and the consistency of views on system's topology with a factual one. The average of both values becomes a final integrity grade.

The evaluation has been performed using 3-level grade scale, assuming that (1) is the highest note, (3) is the medium and (5) is the lowest one. The table 6 demonstrates the sum of grades assigned to each of the distinguished types of overlay networks.

Table 6. The evaluation of overlay networks dependability

Overlay network	Σ
centralized topology, centralized directory model	19
hybrid topology, flooded requests model	17
hybrid topology, resource routing model	13
decentralized topology, flooded requests model	15
decentralized topology, resource routing model	11

As it might be expected, the most dependable overlay network is that using decentralized topology and resource routing model, as opposed to centralized topology with centralized directory model. The most compromised model is that linking a hybrid topology with a resource routing model.

The overlay networks using resource routing models were developed as an improvement of those based on flooded requests model and, indeed, received higher grades during the analysis.

We can conclude that the topology used in an overlay network is a determinant of reliability and adaptability. It means that the number of centralized points in a system imposes the reliability and adaptability level.

Regarding availability, the choice of a resource discovery algorithm is crucial. So, the most promising overlay networks are those making use of a centralized directory model. The reason is that this approach engages the lowest number of queries during the resource discovery process. Moreover, the response time is the shortest (assuming the central point is not overloaded). The alternatives are systems based on flooded requests model due to their unstructured search procedures, which entails a huge overhead. Even if the overhead is reduced by using time-to-live parameter, it additionally diminishes resource availability (negative reply may be obtained even if the resource is available). An improvement of flooded requests model is a resource routing model. It enables structured searching, hence the number of queries is noticeably lower.

With respect to scalability, a topology as well as a resource discovery algorithm is significant. The centralized topology is the worst choice as one centralized point creates an evident bottleneck. The visible influence of resource discovery model appears in flooded requests model case. As the number of peers increases, also the number of propagated queries does, which directly lengthens a response time.

5. Conclusion

The analysis of overlay networks with respect to their dependability has been carried out for five types of overlay networks, selected on the basis of their topologies and resource discovery algorithms.

It has been shown that the best overlay network according to its dependability is that with decentralized topology supported by a resource routing model. Slightly less satisfactory networks are mixtures of hybrid topology with a resource routing model and decentralized topology with a flooded requests model.

Regarding the results of the performed analysis as well as usefullness of abovemenioned systems for software development, JXTA turned out to be the most appropriate choice. It is almost as good as the most dependable solution and, moreover, it is the most convenient for further improvements (as this is neither a protocol nor a concept, but a ready to use library). It is feasible to extend its dependability by adding availability-consistency tradeoff.

Future work will be focused on development and implementation of a prototype system based on JXTA, extended by availability-consistency tradeoff.

10. References

[1] A. Avizienis, J.C. Laprie, and B. Randell, "*Fundamental Concepts of Computer System Dependability",* IARP/IEEE-RAS Workshop on Robot Dependability: Technological Challenge of Dependable

Robots in Human Environments – Seoul, Korea, May 21-22, 2001
[2] L. Melville, J. Walkerdine, and I. Sommerville, "*Dependability properties of P2P architectures*", Proceedings of IEEE P2P 2002, Sweden, 5th-7th September 2002
[3] IFIP WG10.4 on Dependable Computing and Fault Tolerance, http://www.dependability.org/wg10.4/
[4] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu, "*Peer-to-peer Computing by HP"*, available at www.hpl.hp.com/techreports/2002/
[5] D. Werthimer, J. Cobb, M.t Lebofsky, D. Anderson, E. Korpela, *"SETI@home: an experiment in public-resource",* Communications of the ACM Volume 45, Issue 11 (November 2002)
[6] website http://setiathome.ssl.berkeley.edu
[7] website http://www.kazaa.com
[8] J. Liang, R. Kumar, K.W. Ross, "*Understanding KaZaA*', available at http://cis.poly.edu/~ross/
[9] B. Traversat, A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J. Hugly, E. Pouyoul, B. Yeager, *"Project JXTA 2.0 Super-Peer Virtual Network"*, available at http://www.jxta.org/project/www/docs/
[10] H. Balakrishnan, M.F. Kaashoek, D. Karger, R. Morris, I. Stoica, "*Looking up data in P2P systems",* Communication of the ACM, February 2003, vol. 46, No 2
[11] P. Maymoukow, D Mazieres, "*Kademlia: A Peer-to-peer Information System Based on the XOR Metric",* Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), 7-8 March 2002 - MIT Faculty Club, Cambridge, MA, USA.
[12] J.E. Berkes, "*Decentralized Peer-to-Peer Network Architecture: Gnutella and Freenet",* available at http://home.cc.umanitoba.ca/~umberkes/academic.html
[13] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, "*Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*" In the Proceedings of the ACM SIGCOMM'01 Conference, San Diego, California, August 2001
[14] website http://www.napster.com