

Zone Based Peer-to-Peer

Rüdiger Schollmeier

Lehrstuhl für Kommunikationsnetze
Technische Universität München
Munich, Germany
Schollmeier@ei.tum.de

Wolfgang Kellerer

DoCoMo Communications Laboratories Europe GmbH
Future Networking Lab
Munich, Germany
Kellerer@docomolab-euro.com

Abstract— Peer-to-Peer (P2P) networks without central entities, such as Gnutella or JXTA, generally suffer under a high signaling load resulting in poor efficiency. The main reason therefore is the necessity to flood requests in the overlay, since in most P2P protocols the nodes are not provided with any information about the P2P overlay network topology. This paper therefore addresses this application-layer routing problem, by proposing a new P2P routing protocol, the Zone Based P2P protocol (ZBP). ZBP establishes a zone for every peer. In its zone the peer knows the complete P2P overlay network topology and the available content. If a requested content is not available in its zone, bordercast messages are used, to search for the content in neighboring zones. Employing these concepts, ZBP achieves a notably improved signaling performance, compared to completely other routing approaches such as Gnutella 0.4 or Gnutella 0.6. As a proof of concept, we analyze the signaling performance of ZBP nodes and Gnutella nodes, with means of random graph theory and ns-2 simulation.

Keywords— Peer-to-Peer, signaling, traffic, replication rate, application layer routing, resource sharing, Random Graphs

I. INTRODUCTION

P2P networks were initially developed for P2P file-sharing, based on TCP/IP networks. The significance of distributed information sharing systems has been demonstrated through the popularity of P2P applications such as Gnutella 0.4 and Napster, since 1999. Recently, also other applications employ the benefits of P2P networking, such as P2P Media Streaming, Voice over P2P or to provide location based services in Mobile Ad Hoc Networks [1].

The increasing popularity of P2P networks can also be observed in the impacts of P2P traffic on common IP networks. In the Abilene backbone, between 10% to 50% of the total traffic is caused by P2P applications [16]. Further on the ISP of the Technische Universität München reports an increasing symmetry in the traffic from and to the US. In some networks P2P traffic already outweighs normal web traffic. In the backbone of the Deutsche Forschungsnetz (DFN) [18] P2P applications sometimes already cause up to 70% of the total traffic volume [17].

Having a closer look at the traffic statistics of P2P networks, it must be stated, that most of this traffic, which is up to 1.381 TByte [16], is caused only by signaling messages within the P2P overlay network. The signaling messages are used to search for resources and to guarantee sufficient

connectivity in the P2P network. The current flooding strategy employed in pure (Gnutella 0.4) and parts of hybrid P2P networks (Gnutella 0.6)[3] limits the signaling efficiency of P2P networks. The nodes have no knowledge which peer contains the requested information and how the overlay network is set up in its proximity. Therefore, a peer needs to broadcast messages on the overlay network to detect which nodes are currently available. Broadcasts are propagated to every direct neighbor of a peer. Likewise, upon receiving a query, these peers propagate the query to all of their neighbors, which then distribute it again in a similar manner. Obviously, this model is not efficient because of the large signaling traffic volumes, compared to the transmitted user data.

Many architectures and algorithms have been proposed to solve the inefficiency problems of Peer-to-Peer networks. Most of them tend to establish hierarchies in the P2P network. Gnutella 0.6 [3], e.g., establishes a two-tier concept, with so called Ultrapeers in the higher hierarchy level and so called leafnodes in the lower level. However, in the higher hierarchy flooding is still employed to search for content and to provide connectivity. This leads again to high traffic volumes as shown in [19] and [2].

In this paper, we propose Zone Based Peer-to-Peer (ZBP) as a novel P2P architecture, which aims to reduce broadcasting in the overlay network to a minimum, by introducing limited, dynamic knowledge keeping of the overlay network topology in a distributed manner in each of the peers. With this approach we provide a suitable architecture for an efficient search mechanism by additionally keeping the symmetric Peer-to-Peer character of the network.

Our concept improves the efficiency of a pure Peer-to-Peer network by combining reactive routing with proactive routing for fast and traffic-efficient location of requested content. ZBP employs no centralized entities. No hierarchies are imposed to the nodes participating in the network. ZBP allows participating members to share all kinds of resources in a pure Peer-to-Peer overlay network. Shared resources can be computing power, cryptographic keys, content files of any type, meta-information on describable resources and any other kind of service.

The remainder of this paper is structured as follows. First, we discuss existing P2P protocols in Section II. Section III provides a description of the general architecture of ZBP and Section IV describes in detail the specific protocols of the ZBP-stack, in particular the Zone Setup Protocol (ZSP), the Query Routing Protocol (QRP) and the Border Resolution

Protocol (BRP). Before concluding this work in Section VI, we present analytical evaluation and ns-2 based simulation results of ZBP, in Section V.

II. RELATED WORK

Unstructured P2P networks can be grouped into centralized, pure and hybrid P2P networks [13]. Centralized P2P networks, such as Napster [14][15], are characterized by a central lookup server, to which the peers direct their requests. In contrast, pure P2P networks, like Gnutella 0.4, omit central entities, and their requests are simply flooded within the overlay network, leading to high traffic volumes. Hybrid P2P networks, such as Gnutella 0.6 or JXTA, implement a second, dynamic hierarchy level, to avoid flooding on every peer. Nodes in the lower routing layer (leafnodes) direct their requests to the search hub (Ultrappeer in Gnutella 0.6, Rendezvous Peer in JXTA), which then broadcast this request in the higher hierarchy of the overlay network. This approach decreases the overall signaling load by the cost of introducing asymmetric signaling [19]. However the load on a hub increases linearly with the number of leafnodes, which thus limits the number of leafnodes per hub. Independent from their architecture, i.e. centralized, hybrid or pure P2P, most of the P2P protocols are usually based on reactive routing schemes. *Reactive* routing in this context means, that a route from the querying node to the node providing the requested resource is only developed with means of route request messages, when the user initiates the request.

The nodes participating in such a network possess only very limited knowledge about the overlay network topology of their proximity. They only know, which P2P nodes are available in their proximity to send a request as part of the flooding scheme, but they do not have any knowledge about the shared content, or on which overlay path a node hosting the content can be reached. Also, for keep alive issues in traditional systems peers only know some arbitrarily selected nodes, which are currently connected to the P2P overlay network, and how they could connect to them on the transport layer.

P2P systems based on Distributed Hash Tables (DHT), like CAN [6], Chord [7], or Tapestry [5], can be regarded as a first step towards *proactive* routing. In DHT-based overlay networks the path to certain content is set *before* a request for content is issued by the user. Peers and the hosted content are labeled by hash keys. To make sure that content can be found by using the hash key as an indication for a route to the requested content, every new content, brought into the overlay, is transferred to that peer with the minimum distance between the hash value of the content and the node ID. Since content is moved in these P2P networks to facilitate routing, we refer to these systems as structured P2P networks.

Nodes in a DHT based P2P overlay network connect to each other, depending on their hash key, i.e., the overlay network is built in way that a node connects always to that node, with the minimum difference according to their hash values. Assuming, that every node has only two neighbors, the network is established as a virtual, ordered chain. Thus routing to nodes is done by simply sending the request in the direction of increasing hash values, as long as the hash value of the request is smaller than the ID of the node receiving the request.

This predetermination of the route before a particular request is issued can be regarded as proactive routing.

The significant decrease of routing overhead in DHT based P2P networks comes at the cost of requirements, which may not be acceptable for every application. In DHT based networks, any content or at least a link to that content brought into the network by participating nodes has to be transferred to that node with the minimal difference between the describing hash values. This may not be applicable to all applications and might also lead to scalability problems [25]. Further on, any content can only be located by its globally unique ID and all replicas described by one keyword are stored on one node. This presumably leads to a small number of peers which have to store an excessive number of objects, as certain keywords are commonly associated to a large number of files [26].

Except in structured P2P networks like CAN or CHORD, which also have some drawbacks as described above, proactive routing schemes are currently not employed in P2P networks to our knowledge. However proactive routing schemes are employed successfully in Mobile Ad Hoc Networks, like in DSDV [20], AODV [21] or ZRP [22].

The Zone Based Peer-to-Peer (ZBP) routing approach described in this work is based on the idea of employing proactive routing within a certain zone and reactive routing outside of this zone. In contrast to ZRP it is completely independent of the physical layer and provides possibilities to route for any kind of object, not only addresses of nodes. In this paper, ZBP is based on TCP/IP as a transport layer, but it can certainly also be employed on any other transport protocol.

III. ARCHITECTURE

In contrast to Mobile Ad Hoc Networks, the employment of proactive routing in the complete P2P network is neither sensible, nor feasible, as the number of participants in P2P networks is magnitudes higher, than in MANETs [3]. Further on we assume a certain replication rate of each object available in the Peer-to-Peer network. Thus it is not necessary to search for an object in a large area, as it can be assumed, that the requested object can be found within a certain proximity of the requesting node. Thus the basic idea of ZBP is to employ a proactive routing algorithm within the zone and reactive routing outside the zone. The size of the zone depends on the availability of the data, i.e., the average replication rate. The replication rate can be assumed in P2P networks to be 0.0055 [23], resulting in an optimal zone radius of 2 to 3 hops in the overlay network, which we prove in our analytical evaluation in Section V.

A. Structure

As illustrated in Figure 1, ZBP consists of three protocols, the application interface ZBP, the Zone Setup Protocol (ZSP), the Zone Query Protocol (ZQP) and the Bordercast Resolution Protocol (BRP). Additionally, HTTP is employed in ZBP to provide the nodes with data exchange functionalities. ZBP receives user inputs, controls and coordinates the ZSP, the ZQP and HTTP accordingly. ZSP is responsible for the management of the zone of the node, i.e., it sends out peer advertisements and handles incoming peer advertisements. The ZQP instance provides the node itself with the search functionality. Further on ZQP provides the node with routing functionalities to

handle incoming Bordercast messages from other nodes. Bordercast requests are handled by the Bordercast Resolution Protocol (BRP).

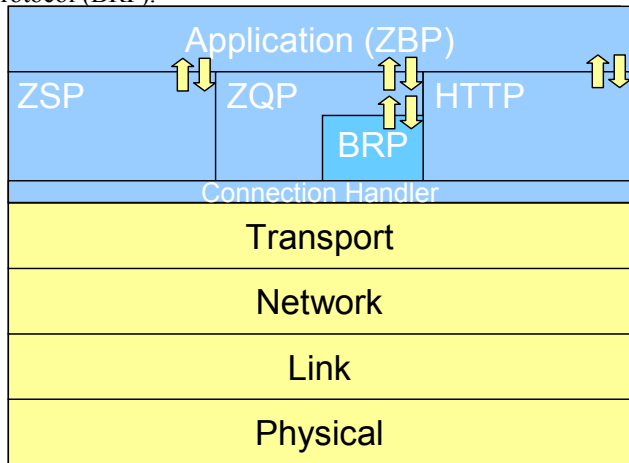


Figure 1 Protocol Stack of ZBP

The connection handler finally offers an interface between the ZBP and the transport layer, and is currently designed as an interface to the TCP/IP layer. It offers configuration possibilities, to influence the way connections are established. For example, the overlay network connections can thus be adapted to the physical layer topology to minimize connection length. It works as a kind of cross layer communication channel, as it can retrieve information about hop distances or delays also from the transport layer. Thus the connection handler can establish connections, according to criteria and parameters from the application as well as from the transport layer.

Every peer is the center of its zone. A zone is defined by the zone radius, which we set in the following to a value of 2. This means, that every node which can be reached within 2 hops (one hop is the link between two nodes in the overlay network), is the member of the zone of the new node. For a better understanding the zones for node 5, node 20 and node 15 are illustrated in Figure 2. As every node establishes its own zone, the zones overlap as depicted in Figure 2, offering a smooth transition of the zones and eliminating border effects.

Within each zone, the center node sends its announcements to its zone members. In case of zone 6 in the example of Figure 2, node 6 is the center and the nodes 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13 are its zone members. As the center node is also a member of the other zones, it also receives their announcements and thus knows exactly which data is available in its zone and where it can download the data from. If the data is not available in its zone, the center node directs a request to its border nodes, which look up their tables to find the content in their zone. In the example, depicted in Figure 2, node 2, 20, 8, 12 and 13 are border nodes of node 6.

In the case, that the requested content can not be located in their zone, the border node forwards this request to its border nodes, but not the one it received the request from. Forwarding the request to its border nodes will continue, until a predefined number of forwards is reached, or the demanded content is available in the queried zone. In the case of a success a

response, containing all necessary information, e.g., IP address and download path response, is routed back to the node which initiated the query. Thus the node has all information necessary to download the requested data from the node specified in the response message via HTTP.

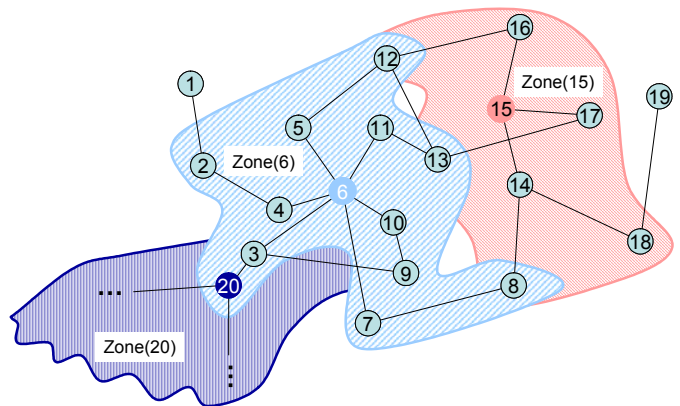


Figure 2 Examples of Zones established by the ZSP (zone radius 2)

B. Behavior

On startup, ZBP first has to connect to at least one active node, participating in the ZBP overlay network. Therefore the node has to try to connect to nodes, it knows from previous session, stored in a cache. If no valid addresses are available, the node has to contact a beacon server to receive valid addresses of active ZBP nodes, similar as in Gnutella. If enough addresses are available, the ZBP instance triggers its connection handler to establish the according connections.

The connection handler first establishes a connection on the transport layer, e.g., a TCP connection. Then it exchanges a handshake with the active ZBP node to validate the connection. This connection establishment process is illustrated in Figure 4, for the first 9 messages. Figure 4 gives the message sequence chart for the case, that node 2 to node 8 are already active participants of the ZBP network, and node 1 wants to participate as a new node in the ZBP network. The resulting zone for node 1 is shown in Figure 3. Thus node 1 becomes a member of the virtual network, although it only knows its direct neighbors (node 2, node 3 and node 8). However it does not yet know what content is available in its zone.

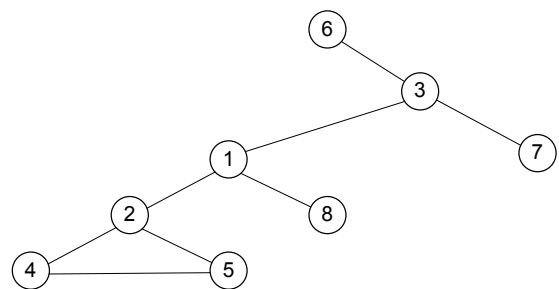


Figure 3 Example ZBP zone (zone radius 2) for node 1

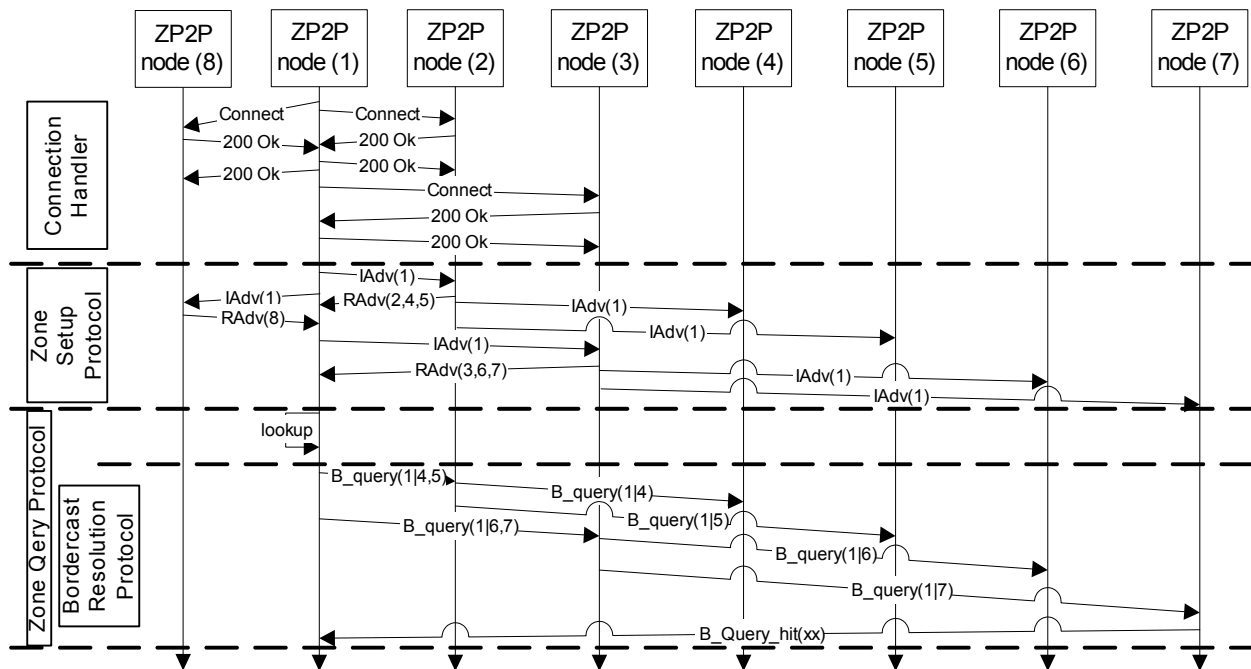


Figure 4 Message sequence chart for connection establishment and the data search process in ZBP network

To announce its presence and to receive information about the content available in its zone, node 1 distributes an advertisement (IADV) in its zone. On receiving the advertisement from node 1, the direct neighbors of node 1 (node 2, node 3 and node 8) respond with their advertisement (RADV) and additionally transfer the initial advertisement (IADV) to the zone members, which are more distant members of the zone of node 1. An RADV includes already the advertisement of the peers which are more than one hop away from node 1, due to previous advertisements they have received within their zones. Thus further transfers of advertisements via additional links can be avoided (see Figure 4). Consequently node 1, as the center node of zone 1, knows now about all content shared in its zone and the topology of the network. Additionally, all zone members of zone 1 know the content shared by node 1. To keep the tables about the shared content up to date, the ZSP of each node distributes in its zone incremental update messages, whenever the shared content of the node changes.

If the user of node 1 searches for content in the ZBP network, the request is directed from the ZBP, to the ZQP instance. ZQP then first checks the tables, established from the advertisements the node collected with the ZSP. In the case, that the requested content is shared by any of its zone members, it can download the content directly from the node specified in its local routing tables. As illustrated in Figure 5, the probability, that the content is available in the zone, depends on the number of nodes within the zone, and thus on the zone radius of the zone. With a zone radius of 2 hops the probability, that the content can be found within the zone, reaches already an acceptable value, if we assume an exponential distribution of the nodes' degree (as will be shown in Section V). However, in the case, that the content is not available in the zone, ZQP sends a request command to the BRP, which sends a bordercast request message to the zone's

border nodes (B_QUERY) (see Figure 4). As mentioned above, the border nodes themselves look up their tables for the requested content and forward the request to their border nodes if necessary. Figure 4 illustrates the case, that the content is not shared within the zone, but shared by one of the neighboring zones. In case that the content is available in a zone, which received a B_QUERY, the according response is transferred to the requesting peer. In the example above, the content is available in the zone of node 7, which sends back B_QUERY_HIT message (see Figure 4).

BRP signals this result to ZQP, which notifies ZBP about the availability of the requested data. Thus the user can initiate a download via HTTP from the providing node. As soon as the download is successfully completed, an incremental update message is distributed within the zone, to notify the zone members about the new content available in the zone.

IV. PROTOCOLS

For the exchange of messages between the peers ZBP uses a general message header, which includes a Node Global ID (NGUID), a message type field, the Zone Radius, a hop counter field and a Payload Length field. The NGUID uniquely identifies the nodes in the ZBP network. It is a 16 byte string which can be constructed by applying a MD5 hash to a string concatenated from the IP address of the node, the date and time of the installation of the user software. The message type field indicates the type of the message and therefore defines the structure of the payload.

The hop counter field is used, to count the hops, how far a respective message has been forwarded already. This value is incremented at any peer forwarding this message and the receiving peer always compares this value to the value given in the zone radius field. If both values match, the message is deleted and not forwarded any further within the network. The Payload Length field gives the number of bytes of the payload,

which includes, e.g., the content of the advertisements, i.e., the description of the shared files depending on the message type, the Route field, which describes the route a message has to take (e.g., for the RADV, see below), or the path a message has already taken (e.g., for the IADV, see below). The length of the Route field is accordingly defined by the hop counter or the zone radius. If we specify a fixed zone radius for each node, we could reduce the message size even further, because we can omit the route information of all messages completely, as any node within a zone knows the complete topology of the zone.

As outlined in the description of the general structure and behavior of ZBP in Section III, the Zone Setup protocol (ZSP) is responsible to provide connectivity and to collect, build up and distribute the tables for the Zone-Query Protocol. It defines 4 types of messages, the Initial Advertisement (IADV()), the Response Advertisement (RADV()), the Add Advertisement (AADV()) and the Eliminate Advertisement (EADV()). The Route field of ZSP messages includes in the case of an initial advertisement the route the message has traveled so far. This means that every peer forwarding the IADV() adds its NGUID to the Path field. Thus response and update messages, like the RADV() or the AADV() can be routed within the network along the shortest path in the overlay network. This route is stated in the Route field of update and response messages.

An advertisement of one shared file consists of the locally unique description of the resource, e.g., a filename, an MD5 hash value of one globally unique descriptor of the service, e.g., the MD5 hash value of the data-file, and 4 hash values of four keywords as meta-data, describing the shared content. An IADV(), sent to the zone members by a new center node, includes the information about all shared data of the new center node. Upon receiving the IADV(), only the direct neighbors of the new center node answer with a RADV(), containing all of the information about the data shared by themselves and the data of further zone members, defined by the zone radius, stated in the IADV().

To keep the information about the shared content within one zone up to date, incremental updates to either remove (EADV()) or to add (AADV()) information about the shared content, are exchanged between the members of one zone. To keep the amount of exchanged data as small as possible, incremental updates are employed. They contain only the information about the files which either have to be removed or added to the shared list. An AADV() must always be sent, as soon as one file is successfully downloaded and shared. As it can be assumed, that every peer stays in the network for about 10 minutes and successfully downloads at least one file [19], we can additionally employ the incremental update messages as keep alive messages, so that all other zone members can be sure about the existence of the peer. If no advertisements are received from one node within 10 minutes, the peer is assumed to be no longer an active member of the ZBP. It will therefore be removed from the routing tables.

Based on the routing tables provided by the ZSP, the Zone-Query-Protocol (ZQP) is used to establish routes to objects requested by the user via the ZBP-interface. In the following we distinguish three possible states of a node:

- Center node: node initiating a request as a center node of its zone

- Border node: node located at the border of a zone, receiving a border query from its center node
- Inner node: node which is neither the center node, nor a border node and therefore has to route request-messages (B_QUERY()) from the center to the border node or response-messages (B_QUERY_HIT()) from the border to the center.

As a center node, the ZQP receives a request for a certain object, from the ZBP initiated by a user request. As a first step, ZQP parses its local routing tables for the requested resource. If it can be found in its local tables, and thus in its zone, ZQP signals the address of the providing node to the ZBP. ZBP can thus initiate a download via HTTP, as described below.

If ZBP can not locate any description of a resource, matching the provided search criteria, ZBP forwards this request to the Bordercast Resolution Protocol (BRP). The BRP sends this request to the zone's border-nodes via a Borderquery-message (B_QUERY()). The B_QUERY()-message carries as payload the hashed search keywords, ZQP received from ZBP. Further on it contains a zone counter, to count the number of traversed zones, and the maximum number of zones, the B_QUERY() must be forwarded. Additionally, ZBP adds the route to the border node, so that the message can be routed by the inner nodes, to avoid unnecessary message overhead. The inner nodes increase the hop counter and decrease the path length by one and additionally remove their ID from the path list, before they forward the message to the next node from the path list. Thus the message can be decreased hop by hop, to save further bandwidth. To be able to route a possible result message (B_QUERY_HIT()) back to the initiating center node, every inner node stores for a preconfigured amount of time the NGUID of the message and the ID of the node it received the message from, to provide backward routing capabilities.

As soon as a border node receives a B_QUERY(), it compares the hashed search keywords of the payload, with its own routing tables. If the search keywords matches the description of a content available in the zone of the border node, the border node sends back a B_QUERY_HIT()-message. This message contains the port number and IP-address of the node providing the requested content, a unique description of the requested resource, e.g., MD5 hash-key of a file, and a description, how the object can be accessed, e.g. the filename. The MD5 hash-key is in this case used, to enable the requesting peer to continue a download from another source if one source goes offline, or even to download the object from several sources at the same time. When the BRP instance, which initiated the B_QUERY(), receives a B_QUERY_HIT(), this message is forwarded to the ZBP via the ZQP. Upon further user interaction, the ZBP instance can initiate a download via HTTP.

If the border node does not find a matching resource in its own tables, it sends a request via broadcasting. This means, here our concept switches from proactive routing to reactive routing as we know from other P2P concepts.

ZBP utilizes HTTP/1.1 [4] for data transfers between peers. The implementation of the required HTTP clients and HTTP servers must be RFC compliant. Additionally ZBP specifies the behavior pattern of servers and clients as far as it is not covered by HTTP, e.g., the case of connection breaks.

As any node might leave the network at any time in P2P networks, especially during long transfers, ZBP employs the content range header of HTTP. Thus it is possible to continue the transfer from the last received byte on. Additionally the content ranger header offers the possibility to transfer a file in several parts and from several sources in parallel to stabilize and speed up the transfer in ZBP.

To enable uploads of files, ZBP also implements the HTTP PUT request method. PUT is compliant to the HTTP RFC, but not mandatory for HTTP servers. In ZBP PUT is mandatory, to guarantee full upload functionality.

V. RESULTS

As a proof of concept we specified ZBP in the System Definition Language (SDL) based on the Telelogic simulation tool. With this kind of prototype we can test the developed protocols against their requirements and can demonstrate the basic functionality of the system. Furthermore, we use it as a detailed protocol specification.

The SDL simulation specifies a network of eight nodes, which can be connected to each other via a connection manager, also implemented in SDL. Via the configuration manager we can additionally set the zone radius of each node and can deploy content on the nodes. Further on we can also initiate search requests and advertisements from any node, leading eventually to Borderquery-messages. Thus we can visualize the message flows between the different instances and nodes and can verify the behavior according to the protocol requirements described in this work.

To be able to evaluate the signaling efficiency of ZBP in comparison to other P2P protocols, we evaluate ZBP and the Gnutella protocol analytically. Therefore, we employ an approach based on random graph theory. We base our analysis on findings published in [10] and [11]. As described in [11] we base the computation of the number of nodes available in the random graph, on the mathematical concept of generating functions.

As general assumptions for the random graph, we use an exponential distribution of the node degrees in the application layer, so called virtual vertices, as given in (1). From previous measurements we assume an average degree of $\mu=3.0$ connections per node [24], resulting in a variance of $\sigma=3.46$ and for the coefficient of the exponential distribution $\kappa=3.48$. With the analytic concepts described above we can thus compute the number of reachable nodes, as seen from one node against the number of hops. We also evaluated the protocols with other degree distributions, e.g., a truncated Powerlaw distribution, with similar results. Due to the limited number of pages we do not cover these models in further details.

$$p_d = \left(1 - e^{-d/\kappa}\right) e^{-d/\kappa} \quad (1)$$

For the behavior of the nodes in the P2P network we assume an average uptime of 900 seconds [19], an average of 100 files shared by each peer and two downloads per session. Further on we assume an average replication rate of the data in the network of 0.55% [23]. Replication rate in this context means, that a certain object is shared with the probability of 0.55% on a certain node of the P2P network. With these assumptions, we want to characterize a real P2P network as far as possible. However, the above stated values do not have any

effect on the comparison, as they are the basis for both analyses the Gnutella analysis and the ZBP analysis. Our analysis is restricted on the application layer traffic, i.e., we do not take into account any additional header, e.g., TCP/IP headers, or aggregation effects of the transport layer and further lower layers. We assume the same transport layer for both P2P systems and thus can exclude any effects on our comparison of both P2P systems.

From the ZBP protocol specification, we calculate a size for a single update message of 37 byte, resulting in 3.7kbyte for the initial advertisement (IADV()) of a node (100 files). For the general header length of ZBP we have 23 byte, for the size of the B_QUERY()-message we 80 byte and for a B_QUERY_HIT()-message we 200 byte. The zone radius in our analysis varies from 2 to 7 hops. With the replication rate we can compute the average availability of a specific content within the zone and within all neighboring zones (see (1)). This results in an availability graph, as depicted in Figure 5. Here we can observe that in very small zones, the content availability is low. However, it increases very fast to a value of 1, which is reached already, when the zone-radius is 4. If the network can cope with additional B_QUERY()-messages, and thus allows the node to search for content also in directly neighboring zones, an availability of 1 is already reached with a zone radius of 2.

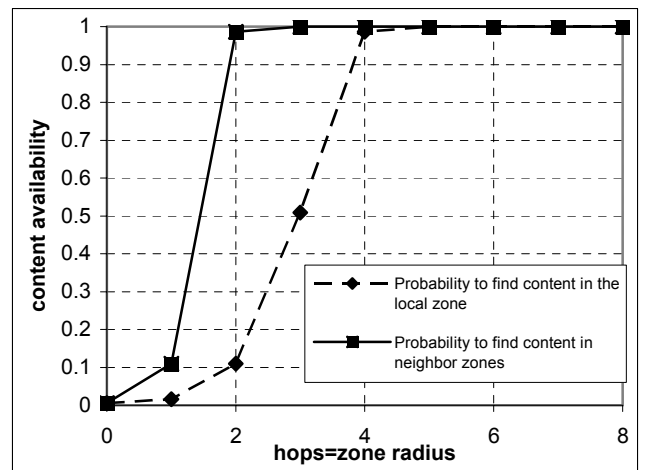


Figure 5 Availability of a specific content in average, against zone radius for $\kappa=3.48$ (dashed: availability of the content within the zone, solid: availability of the content in the inner and all neighbor-zones)

Concerning Gnutella we analyze the Gnutella 0.4 protocol, with a general header length of 23 byte, 0 byte for the PING-message, 15 byte for the PONG message, 80 byte for the QUERY message and 200 byte for the QUERY-HIT message. As Gnutella routing is based mostly on flooding, we additionally have to take loops in the network into account, on which a message may be transmitted twice via the same link. Therefore we assume a loop probability of 0.0064 [24], i.e., that a node is a member of a loop and thus receives one message twice.

The traffic emitted and received by one node depends on the size of the component a node is a member of. To compute the size of the component, i.e., the number of nodes reachable within the hop distance h , we use the concept described in [11]. With these numbers, our assumptions stated above and the

knowledge of the protocol, we can thus compute the signaling traffic caused and received by each node.

We sum up the traffic caused by all messages for a Gnutella node or a ZBP node, respectively, over the average lifetime of one node (900 seconds), because only for this time we can assume a stable virtual network topology in average. After 900 seconds the topology is changed, because of leaving and new joining nodes, which would result again in the traffic stated above, and thus this does not influence the total data rate of one node. Thus we can compute the average data rates for every node, as depicted by Figure 6. In Figure 6, two hops correspond to one increment in the zone radius, because we allow in this analysis searches also in directly neighboring zones.

As depicted by Figure 6, ZBP outperforms Gnutella considerably concerning the signaling traffic. Further on ZBP additionally increases the search performance, as already the content availability is very high within one zone, if we choose a zone radius of 4. If we allow B_QUERY()-messages as it is assumed in this analysis, an average content availability of nearly 100% can already be achieved with a zone radius of 3 (see Figure 5). A zone radius of 3, results in a content availability within the zone of 51% and with a B_QUERY of 100%. This means that more than 50% of all requests of the user can be satisfied instantly, as a providing source is already stated in the nodes tables, generated from the received advertisements. In the case, that the content is not available in its zone, then the content can be located with a hop-by-hop, route-able B_QUERY()-messages sent to its border nodes.

Concerning Gnutella, it would always be necessary to broadcast a QUERY() across at least 5 hops, to be sure to find the content, as indicated by the dashed line in Figure 5. Broadcasting these messages, which is completely avoided in ZBP, therefore results in notably higher data rates per node on the receiving, as well as on the transmitting part. Even if we compare ZBP to hierarchical approaches, like Gnutella 0.6, ZBP is still better in its traffic behavior. In Gnutella 0.6 an Ultrapeer with 25 leafnodes has a send data rate of 169.96 kbit/s and a receive data rate of 150.54 kbit/s, whereas a leafnode has a transmit data rate of 1.07 kbit/s and receive data rate of 1.20 kbit/s [19]. The average transmit and receive data rates of a Gnutella 0.6 node, thus result in 7.56 kbit/s and 6.94kbit/s respectively, if we assume 25 leafnodes being connected to one Ultrapeer. In contrast, a ZBP node, with a zone radius of 3 has an average transmit data rate of 0.89 kbit/s and an average receive data rate of 4.45 kbit/s. On the one hand these data-rates are significantly smaller, than of an average Gnutella 0.6 node, and additionally ZBP avoids nodes with a high load, like the Ultrapeers in the Gnutella 0.6 network. Further on, the content is available instantly, i.e., in ZBP it is, depending on the zone radius, not necessary to query the network and to wait until the network responds with according QUERY_Hit() messages.

To verify that the results of ZBP are not only based on reduced message sizes, we additionally implemented a Gnutella Protocol which employs string compression for the signaling messages [12]. Thus we can reduce in general the overhead by an average of 40% to 50%. Resulting we assume a common compression factor of 0.6, for the enhanced Gnutella Protocol, depicted by the dash-dot line in Figure 6.

We can observe, that this compression decreases the signaling traffic notably, but it can not reach the performance of ZBP.

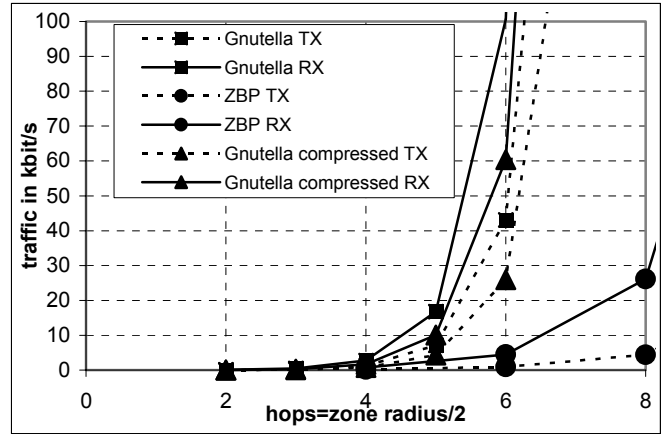


Figure 6 Traffic caused and received by one Gnutella node, one enhanced Gnutella node and one ZBP-node, for $\kappa=3.48$

To validate the results of our theoretical analysis, we performed simulations for Gnutella and ZBP. We used the network simulator ns-2 [9] for our simulations, which include also the transport and the network layer. We simulated topologies with 100, 200, 300 and 700 nodes, with a power law overlay degree distribution, as stated above. Due to the complexity of the simulations, we could not simulate larger networks. This explains the differences between the simulation and the analytical approach, as the basic assumption of the analytic approach are infinite networks, whereas the simulations are restricted to a maximum of 700 nodes. We are currently developing an analytic approach which takes finite networks into account, but are not able to present its results yet.

Analyzing the traffic received and initiated by one ZBP node, we can clearly observe in Figure 7 that in contrast to Gnutella networks, the traffic does not grow with network size, but is constant. The reason therefore is, that a node sends only packets to members of its zone, which is independent from the network size. Thus the average data rate per single node does not increase with a growing network size.

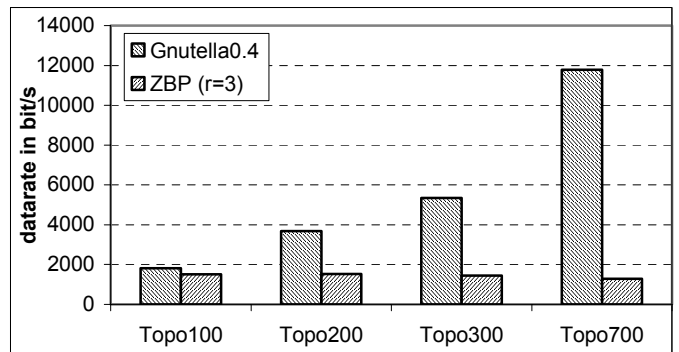


Figure 7 Average Total Traffic simulated for one ZBP- and one Gnutella-node

Further on we can also observe, that the success probability in ZBP is significantly higher, than in a Gnutella network. The reason for this interesting result is from our point of view, that

less query hit messages are lost, as significantly less messages are flooded in the network, resulting in less entries in the backward routing tables, which may time out.

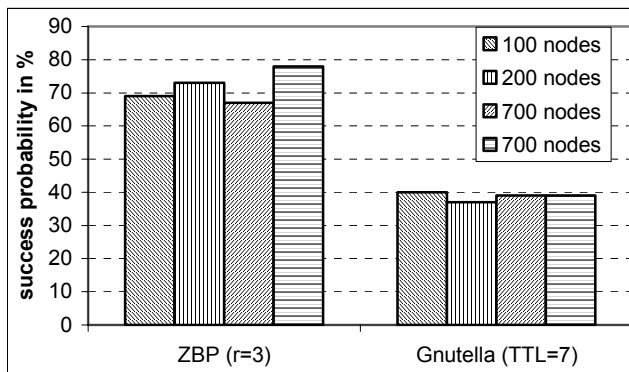


Figure 8 Success probabilities for different network sizes in a ZBP network and a Gnutella network

VI. CONCLUSION

The Zone Based Peer-to-Peer protocol provides a new approach to reduce the signaling load of P2P networks significantly. By increasing the knowledge about the network topology, ZBP limits flooding to a small local proximity of each node, a so called zone. The center node of a zone has the complete knowledge about the topology of its zone and of the content available in this zone. A random graph based analysis and ns-2 simulations of Gnutella and ZBP show that ZBP outperforms in terms of signaling overhead. Even compared to the signaling traffic measurements of hierarchical P2P approaches, ZBP still causes less signaling overhead.

Further on, ZBP allows instantaneous access to shared resources within its zone, as all information of each zone is available on each center node of the according zone, whereas it has to be mentioned, that every peer is the center node of its zone. Especially in conjunction with JXTA, the zone concept of ZBP could be very useful, to support the grouping mechanism of JXTA. Therefore we additionally plan to develop an interface for JXTA in ZBP.

Depending on the zone radius, on the one hand the availability of data and on the other hand the signaling traffic either increases or decreases. The larger the zone, the more content is available, but also the more signaling traffic is caused by each node. Based on our analysis, we would therefore propose a value for the zone radius of three, which guarantees nearly 100% availability and additionally results only in an acceptable total average data rate of 5.3 kbit/s per node.

In the current version of ZBP, the virtual P2P routing layer is completely separated from the transport layer. Thus ZBP can be employed on any transport protocol, although we currently use TCP/IP for transport. ZBP does not take into account any properties of the underlying physical network. However, with the connection handler and further extension fields within ZBP it is possible to map the ZBP network on the physical network. This might be necessary in location sensitive networks with small capacities, such as mobile ad hoc networks.

REFERENCES

- [1] C. E. Perkins editor, "Ad Hoc Networking", Addison-Wesley, 2000.
- [2] R. Schollmeier, G. Kunzmann. "GnuViz - Mapping the Gnutella Network to its Geographical Locations". Special Issue on P2P networking of the PIK Journal to appear in Spring 2003.
- [3] T. Klingberg, R. Manfredi. "Gnutella 0.6 RFC". June 2002, rfc-gnutella.sourceforge.net/draft.txt.
- [4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. "Hypertext Transfer Protocol - HTTP/1.1". RFC 2616, 1999.
- [5] K. Hildrum, J. Kubiawicz, S. Rao, B. Zhao. "Distributed Object Location in a Dynamic Network". Proceedings of the 14th ACM Symposium on Parallel Algorithms and Architectures (SPAA2002), Winnipeg, Canada. 2002.
- [6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker. "A Scalable Content-Addressable Network". Proceedings of the ACM SIGCOMM Conference 2001. San Diego, CA. 2001.
- [7] I. Stoica, R. Morris, D. Karger, M. Kaashoek, H. Balakrishnan. "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications". Proceedings of the ACM SIGCOMM Conference 2001. San Diego, CA. 2001.
- [8] M. Duigou. "JXTA v2.0 Protocols Specification". IETF Internet Draft. 2003
- [9] K. Fall, K. Varadhan. "The ns-2 manual". Technical report. 2002.
- [10] B. Bollobás. Modern Graph Theory. Springer, 1998
- [11] M. E. J. Newman, S. H. Strogatz, D. J. Watts. Random graphs with arbitrary degree distributions and their applications. Phys. Rev. E 64, 026118, (2001)
- [12] D. Salomon. Data Compression. The complete Reference. Springer, New York, 1997.
- [13] R. Schollmeier. "A definition of Peer-to-Peer networking towards a delimitation against classical client server concepts". Proceedings of WATM-Eunice 2001. 2001
- [14] S. Saroui, P. Gummadi, S. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Technical Report #UW-CSE-01-06-02. 13. K. Anderson. Analysis of the Traffic on the Gnutella Network. CSE222 Project report.
- [15] Napster Messages <http://opennap.sourceforge.net/napster.txt>
- [16] Internet2 Netflow, Weekly Reports, Week of 20020218, 20020923, 20030505 <http://netflow.internet2.edu/weekly>, Mai 2003
- [17] G. Foest, R. Paffrath. "Peer-to-Peer (P2P) and beyond". DFN Mitteilungen 58-3. 2002
- [18] Deutsches Forschungsnetz <http://www.dfn.de>. 18.05.02
- [19] R. Schollmeier, A. Dumanois. Peer-to-Peer Traffic Characteristics. EUNICE 2003. Budapest, Hungary. September 2003.
- [20] C Perkins, P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing ({DSDV}) for Mobile Computers. SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications. 1994
- [21] P. Jacquet, P. Muhenthaler, A. Qayyum. Optimized Link State Routing Protocol. Internet Draft, draft-ietf-manet-olsr-*.txt.
- [22] Z. Hass, M. Pearlman, The Zone Routing Protocol (zrp) for Ad Hoc Networks. Internet Draft, MANET Working Group, March 2000
- [23] A. Asvanund, K. Clay, R. Krishnan, M. Smith. An Empirical Analysis of Network Externalities in P2P Music-Sharing Networks. www.heinz.cmu.edu/~mds/p2pe.pdf
- [24] R. Schollmeier, F. Hermann. Topology-Analysis of Pure Peer-to-Peer Networks. In Proceedings of the Fachtagung "Kommunikation in Verteilten Systemen" (KiVS 2003). Leipzig, Germany, February 26-28, 2003
- [25] L. Liu, K.D. Ryu, K. Lee, "Supporting Efficient Keyword-based File Search in Peer-to-Peer File Sharing Systems". IBM Research Report RC23145 (W0403-068). March 2004
- [26] G. Zipf. "Selective studies and the principle of relative frequency in language". Harvard university Press. 1932.