Performance Evaluation of the Mobile Peer-to-Peer Protocol

Ingo Gruber, Rüdiger Schollmeier
Lehrstuhl für Kommunikationsnetze
Technische Universität München
Munich, Germany
{Gruber , Schollmeier}@ei.tum.de

Wolfgang Kellerer
DoCoMo Communications Laboratories Europe GmbH

Munich, Germany
Kellerer@docomolab-euro.com

Abstract— **The utilization of peer-to-peer (P2P) overlay networks on mobile ad hoc networks (MANET) introduces new services and possibilities such as context based routing and location based services (LBS). A simple layering of both networks is inefficient and does not scale well because the virtual P2P overlay network does not match the frequently changing physical network topology of the MANET. The Mobile Peer-to-Peer (MPP) protocol stack offers a very promising concept by introducing a cross-layer communication channel between the physical network layer and the virtual P2P network layer. This reduces significantly the messaging overhead and increases the search success rate, as we can prove in this work. Therefore we describe an implementation of the MPP protocol stack in the Network Simulator 2 (ns-2). With this implementation we prove the significant performance improvements which can be achieved with MPP, to allow P2P networking in mobile ad hoc environments.**

Keywords— overlay networks, content based routing, location based services, cross layer communication, Mobile Ad Hoc Networks, Peer-to-Peer

I. INTRODUCTION

To provide users with communication services in areas without a suitable networking infrastructure and to employ the resources offered by the wireless terminals, a significant number of Mobile Ad Hoc Network (MANET) solutions have been developed so far. Especially layer 3 routing in wireless multi-hop networks is a well addressed research area. These networks are self-organizing wireless systems that do not rely on any kind of fixed network infrastructures. They operate in a true node-to-node or even peer-to-peer fashion. MANETs evolve in a spontaneous manner, connecting several wireless terminals, the peers, via one hop physical connections to each other. Thus MANETs provide a virtual direct communication channel between any of the participating nodes/peers. In addition, each mobile device acts as a relay node in order to forward messages toward their destinations. MANET access technologies which are already today available are, e.g., IEEE 802.11 and Bluetooth. Both standards are able to create and maintain wireless networks without central entities, just relying on the resources provided by the participating terminals, and therewith fit into the ad hoc network paradigm of infrastructure free communication.

In P2P networks a similar phenomenon can be observed. The content and the intelligence of the network, for a long time kept in the core of the network, are slowly migrating to the edges of the network. In these networks each participant acts as a servent. The phrase servent is derived from the first syllable of “serv-er” and the second syllable of “cli-ent”. This expresses the major characteristic of nodes participating in a P2P network, of being a client and a server at the same time. In contrast to MANETs, P2P systems operate on the application layer of the OSI reference model and establish their overlay network independent from the underlying network infrastructure.

Initially, P2P was first discussed in the mid 1990s, and became famous in the late 1990s as the basis for file sharing platforms. The usage of P2P over fixed networks receives increasing attention in the research community. P2P networks are already employed for distributed computing, file sharing, data and voice communication applications, etc. Due to its broad range of applications and its large amount of traffic these P2P applications already cause today, P2P has also become of higher interest for network providers.

Current MANET routing architectures, like AODV [2] or DSR [3] offer only the possibility to establish routes to a given address. However in most cases the user itself only knows meta data, i.e., context, describing the object the user wants to establish a connection to, but not the IP-address or even MAC-address of the object. In some cases a user might not even want to address one single object, but instead a group of objects which offer a similar service, such as an ATM machine, a taxi, or a replica of a certain file. To solve this problem in a fixed network, often centralized solutions, like e.g. search machines for web browsing, are employed. Such a centralized approach is not feasible in MANETs, as a connection to the central lookup service can not be guaranteed. Further on, such a centralized lookup table can hardly be kept up to date due the frequently changing physical topology of the network.

A solution to this problem can be offered by pure P2P networks, such as Freenet or Gnutella 0.4. They offer a completely distributed routing architecture, which provides the means to enable context based routing. In P2P networks initial queries do not consist of addresses, but of keywords describing the desired object, content or service with at least one keyword. Therefore this approach suits very well to solve the above described problem of context based routing in MANETs.

Besides the well-known file sharing applications based on P2P networks, new wireless applications are thus feasible. For example, a user standing at the side of a road, demands a taxi, but can not see any taxi nearby. He could now call the central taxi agency and order a taxi, having to state its current position. The taxi agency, which has to track the current location of its taxis, could then direct the nearest taxi to the user. In our case, if context based routing would be supported by the available MANET, the user could simply send out a request, which would be broadcasted in a multihop manner, via a pre-configured number of hops in its proximity. All participating nodes would forward the request, until a taxi receives the request. The taxi could then reply with an appropriate response message to the requesting node, and could finally also pick up the user. Thus our context based routing scheme allows the utilization of Location-based Services (LBS) without the need for centralized elements. The underlying MANET limits flooding of the search request to the physical local proximity. Additionally the adaptability of mobile P2P in MPP nodes allows the creation of all kinds of search requests. Possible request categories could thus also include bars, restaurants or closest bus stops.

Since P2P networks and MANETs realize a similar concept, i.e. networking without a structured network, but use different system layers for operation, we think that the idea to combine both architectures is promising. However the simple deployment of common P2P protocols on top of MANETs is inefficient and does not scale well. The reason therefore is mainly because the virtual overlay network does not match the frequently changing physical network, which results in unnecessary zigzag routes, as shown in [1]. To solve this problem, the Mobile Peer-to-Peer Protocol (MPP) employs efficient signaling messages and cross layer communication to decrease the signaling overhead as far as possible and to match the virtual P2P topology on the physical topology of the MANET [1].

In this paper we first provide the reader with an overview of the related work in Section II and present a short description of the Mobile Peer-to Peer protocol in Section III and IV. To prove the suitability of our idea, and to be able to compare the signaling efficiency to other approaches we present the simulation setup and the simulation results of MPP in the Sections V and VI. Section VII finally concludes this paper, by summarizing the main results of the simulation. Additionally it provides an outlook to further research tasks in the area of context based routing in mobile ad hoc networks.

II. Related Work

Most P2P networks are based on, and designed for wire line networks. First P2P applications were developed around the year 2000 to provide a basic communication network for file sharing. Examples are the Napster file sharing system and the Gnutella 0.4 protocol. In contrast to Napster, which employs a central lookup table, Gnutella 0.4 relies on a completely flat architecture. Therefore a large amount of signaling for content and topology discovery is based on flooding in the overlay network. This results in a large amount of signaling traffic in Peer-to-Peer networks [15]. Due to scalability problems, Gnutella 0.4 was significantly improved by the introduction of a hierarchical architecture based on hubs, i.e., so called Super or Ultra-peers in Gnutella 0.6.

A considerable further improvement concerning the signaling efficiency could be achieved by the employment of distributed hash tables (DHT). With such an approach a structured P2P network is established by the participating nodes, which can thus decrease the signaling overhead significantly. However, this can only be achieved, if the content can easily be transferred to any other node. Further on DHT-based P2P systems, such as Chord [9], Can [8]or Pastry [7], do not provide any possibility, to map the virtual network to the physical network. However this is from our point of view mandatory to provide P2P in a MANET environment.

Other approaches such as JXTA intend to provide a general interface, or API, for the development of various P2P applications, like Voice over P2P or file sharing, on the same P2P protocol stack. The middleware provided by JXTA thus separates the logical network completely from the physical network, so that any application can be run on any available network infrastructure. As this overhead is caused by the extensive usage of XML in the JXTA middleware is not always feasible, especially in mobile networks due to the limited resources in terms of bandwidth and computing power, an optimized version of JXTA, the so called JXME has been developed recently. JXME's major improvement is the significant decrease of the employed transfer rate by message encoding. However JXME still does not align the virtual to the physical topology, which might result in zigzag routes as described in [1].

Regarding MANETs, we distinguish two basic routing categories, i.e., proactive and reactive routing approaches. The main characteristic of a proactive routing protocol is to maintain consistent, up-to-date routing information among all nodes in the network. For this purpose, each node must maintain one or more tables to store routing information. Thus the route establishment can take place very fast. The disadvantage of these protocols is the necessity for periodic updates of the network topology. They generate additional traffic, which wastes network bandwidth and battery power. Examples of proactive routing protocols are Destination-Sequenced Distance-Vector Routing (DSDV), and Wireless Routing Protocol (WRP).

In contrast, reactive routing schemes determine the route between two nodes in the network on-demand only. They do not need to exchange control data in absence of data traffic. Only in case a reactive routing protocol wants to set up a route to another node, it floods a route request through the network. In case of success, a response message is sent back either from the destination or an intermediate node, which knows the route to the destination by a formerly-made route request. Examples of reactive routing schemes are the Ad-hoc On-Demand Distance Vector Routing (AODV) or the Dynamic Source Routing (DSR).

Summarizing, we can state, that although pure P2P networks and MANETs employ similar networking approaches, a combination of both networks, by simply establishing a P2P network upon a MANET, without providing any interaction between both networks is certainly doomed to fail. Thus only recently some approaches have been developed to provide context based routing, caching or P2P networking in mobile ad hoc networks, which regard the physical topology.

7DS [4] for example employs local broadcast messages to provide an infrastructure for web browsing without a direct connection to the Internet. Therefore every node acts as mobile cache, which is always renewed if a direct connection is available. A similar approach with an improved caching scheme is described in [5].

Besides providing Internet content in MANETs, ORION [6] is, next to MPP, the only system to our knowledge, which aims to provide P2P services in a MANET. ORION therefore provides a general purpose distributed lookup service and an enhanced file transmission scheme to enable file sharing in MANETs. However it still separates the P2P network to a certain extent from the physical network and thus can not achieve the decrease in signaling overhead, as MPP can.

III. ARCHITECTURE

Peer-to-Peer as well as mobile ad-hoc networks do not depend on a reliable infrastructure. Ad hoc networks must cope with moving nodes, whereas P2P clients suddenly appear or disappear. On the other hand P2P applications based on MANETs have remarkable characteristics like survivability, long term reliability, maintenance-freeness and low-cost utilization, as no additional network management hardware is necessary. To overcome the challenges and to guarantee efficient employment, synergies between P2P networks and MANET must be utilized to reduce the administrational effort and to increase the performance and reliability. The MPP protocol suite consists of the MPP protocol as application layer protocol, the Mobile Peer Control Protocol (MPCP) as the interlayer communication protocol and EDSR as the network routing protocol (see Figure 1). These three protocols provide the basic functionality of the system.

Since MANETs already provide routing algorithms, allowing to find network participants by their IP addresses, an additional P2P implementation of this functionality is unnecessary and even degrades the performance. Resulting EDSR is designed to perform the necessary routing tasks on the network layer and supplements MPP. This approach provides valuable advantages compared to a separate treatment of both networks:

- The MANET controls the organization of the network. Thus changes in the topology of the mobile network are taken into account automatically by the P2P network.
- The network layer is responsible for routing and the application controls the data exchange.
- The integration of both networks avoids redundant information requests.
- The interlayer communication of the protocol optimizes the performance, as the overlay network can be adjusted optimally to the physical network.
- The application layer protocol MPP simplifies the implementation of new services

The separation of data exchange and routing tasks allows the reuse of existing protocols like TCP and HTTP. Only for routing tasks MPP must directly incorporate with in the network layer residing EDSR (see Figure 1).

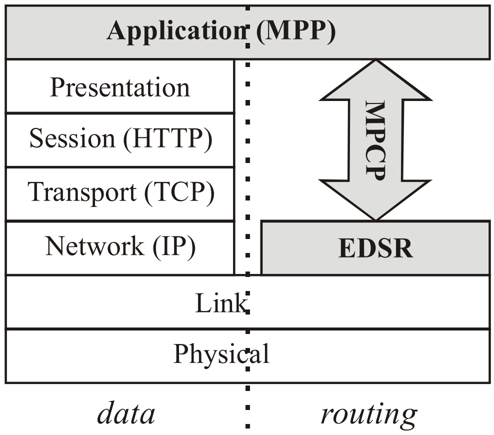

Figure 1: Layered structure of MPP.

MPP allows distant peers to transparently exchange data. Therefore MPP is responsible for file transfers within the P2P network and resides in the P2P client application. MPP utilizes HTTP for data exchange, as it is best suited and well tested. The HTTP content range header is able to resume file transfers in case of network errors due to link breaks. EDSR is mostly based on the DSR protocol, and additionally specifies new request and reply types to provide means to find peers by other criteria than the IP address. EDSR thus extends DSR and therefore EDSR nodes can be an integral part of DSR networks.

MPCP is the interlayer communication channel between the application and the network layer. Thus MPCP links the EDSR Protocol in the network layer with the P2P application in the application layer. Using MPCP the application can register itself in the EDSR layer to initialize search requests and to process incoming search requests from other nodes. It communicates all incoming and outgoing requests and responses to the corresponding protocol (EDSR or MPP), except the file exchange itself.

On startup, the P2P application on the mobile device announces itself to the EDSR layer via MPCP. Figure 2 shows the process of searching and transferring files within the mobile P2P network as a message sequence chart. If a user initializes a data search, MPCP forwards the request to EDSR which transforms it into a search request (SREQ). Similar to DSR route requests (RREQ), EDSR floods SREQs through the MANET. EDSR nodes receiving the request, forward the request to the registered P2P application via MPCP. Thus the P2P application can determine, whether locally shared data satisfies the request's criteria. If the request matches the description of a file shared by the node, the application initializes an EDSR file reply. This reply is sent back to the source node and contains all necessary information for the file transfer. Similar to DSR route replies (RREP), a file reply (FREP) includes the complete path between source and destination.

IV. Protocols

In this section we describe MPP briefly. For more details we refer the interested reader also to [1]. As mentioned above, the general structure of EDSR follows DSR closely to guarantee a seamless integration into the existing DSR protocol (more details in [3]). With these extensions, EDSR offers the possibility to find mobile participants and content, with means of other criteria, than simply the IP address.

To search for data EDSR offers the Search Request (SREQ). It is the initial option for data requests, and is based on the DSR route request. Besides the standard IP-fields, SREQ offers seven additional fields, like an identification field to uniquely identify the message, or a service type field to define the requested service (audio, video, taxi, ATM). Each keyword field of a SREQ holds a 32 Bit hash value of the search string, which represents a search criterion within the predefined service area. At the end of each message the current node stores its own IP address. Thus the route of the reply message is predetermined. Depending on the number of added keywords to the SREQ and the number of already performed forwards, the size of a SREQ can vary between 16 bytes and 256 bytes.

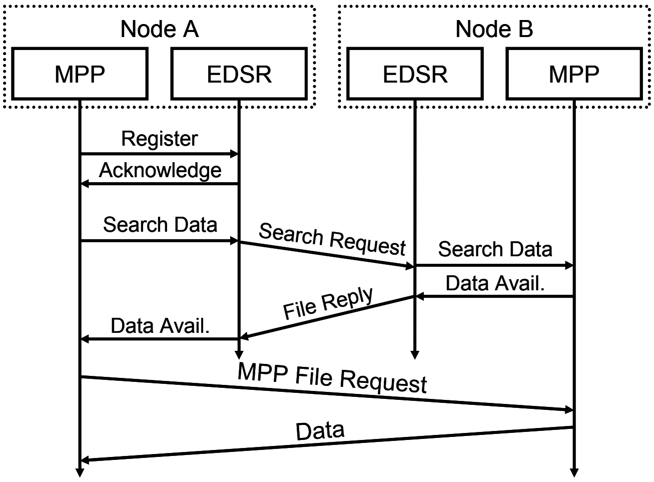

Figure 2: Message sequence chart for data search and download process in the mobile P2P network

The hash request (HREQ) message is a special variant of the SREQ, designed initially for file sharing applications. However HREQ uses only the file size and the 128 bit MD5 hash value (fingerprint) as search criteria. With this specific attribute, HREQ is able to find alternative data sources in case of route breaks as described in more detail below. It is based on the DSR route request as well. The current size of a HREQ depends only on the number of forwarding hops and thus varies from 32 bytes up to 256 bytes.

Peers expect as answers different types of reply packets. The reply type depends on the service and the data-type originally requested. For example, on a received SREQ or HREQ, peers answer with a file reply (FREP), which matches the metadata of the shared object. A FREP implies that the requested data is a file, and therefore includes all necessary information about that shared file. With this information and the help of MPP, the requesting peer can initialize a transmission of the file.

As in the case of search requests, the structure of a FREP is based on the structure of a DSR route reply. Additionally to the existing fields, a FREP contains the file size, the TCP port offered to reach the providing peer, the fingerprint of the file, a bitwise sum of all keywords as matching code of the request and the response and the file name. The size of a FREP can vary from 40 bytes to 256 bytes.

For data transfers between peers MPP utilizes HTTP/1.1 [10]. The implementation of the required HTTP clients and HTTP servers must be RFC compliant. Additionally MPP 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 connection breaks are common in MANETs, MPP employs the content range header of HTTP. Thus it is possible to continue the transfer from the last received byte on.

As depicted in the flowchart given in Figure 2, MPCP, as the interlayer communication protocol, connects MPP and EDSR. It is a synchronous protocol to provide a communication channel between the service layer and the network layer.

Initially MPCP registers all available services with the network layer as one peer possibly offers several services. This enables EDSR to notify the appropriate service about the according incoming search requests. In case the user removes a service, the service has to deregister at the network layer.

MPCP forwards all requests from the EDSR layer to MPP. If the MPP node hosts the requested file or service, MPCP has to communicate the necessary information to the EDSR-layer, which in turn generates the appropriate reply for the requesting peer. In case MPP does not find the appropriate data, MPCP sends a negative acknowledgement back to EDSR. MPCP also transmits search requests with all necessary information to EDSR and receives responses from the EDSR layer.

MPP allows two different handlings of link break events during data exchanges, a static and a dynamic download continuation scheme. Nodes employing the static download procedure cache all replies of a file request for later use. In case EDSR informs MPP about a link break on the primary route, MPP chooses another source from its cached addresses and continues the file download. Similar to the primary route, the selection is based on the shortest path criteria. Only in case all available routes to possible sources show permanent errors, MPP must initiate a new file request.

This scheme reduces the necessary number of search requests as it reuses already gathered information. However this approach also requires the maintenance of a table. Every entry contains information about the requested file, a source, the belonging path to it and the number of hops.

Thus the static download continuation scheme works pretty well for low mobility networks and for small file sizes. However it shows a decline concerning the download performance for networks with fast moving nodes or greater file sizes. Links between nodes break earlier when nodes move with high velocity. Cached information, about possible paths to other sources, becomes stale very quickly. Thus the possibility that secondary paths are still available decreases significantly with the download time. Therewith the number of route error packets, traversing the network, increases while the number of search requests does not decrease.

The dynamic download procedure uses the same initial search request as nodes with the static download procedure. However in contrast to the static scheme, the dynamic download continuation scheme does not maintain a table with all available sharing sources. Initially the MPP client simply chooses the best suited source and discards all other received file replies. It only caches the hash value of the requested file for a possible error handling during the download phase.

In case the path to the source breaks, the requesting MPP client initiates a new request, a HREQ in this case, as it does not have any route to other sources. This HREQ request contains the hash value of the partly downloaded file to clearly identify it. All nodes hosting this file, reply with a FREP packet and the requesting node again chooses the best suited source upon receiving the FREP packets. Afterwards the MPP client can continue downloading the file with the help of the HTTP content range header.

The advantage of this scheme is that the requesting node always uses the most appropriate source for its download. Especially within high mobility networks, this improves the overall network performance significantly, as paths are often shorter and the number of route error packets traversing the network is reduced. Simulations and a detailed analysis to prove these concepts is given in section VI.

V. SIMULATION ENVIRONMENT AND IMPLEMENTATION

This section describes the basic settings and parameters employed for the performance analysis of the MPP protocol suite. For a comparison with other P2P over MANET protocols, we focus on the ORION [6] system, which is the only known system combining P2P and MANET, as described earlier. The performance evaluation is based on the ns-2 simulator. Ns-2 is a discrete, event driven network simulator initially developed by the University California at Berkeley [11]. We perform all simulations using ns-2 version 2.1b9a with the ad hoc network extension from the CMU monarch project [12].

As the MPP protocol stack spans from the network layer to the application layer, we keep a similar structure in the ns-2 implementation. The implementation of the MPP protocol resides in the application layer of ns-2. With the help of a shim layer, the application utilizes the existing TCP agent and the newly developed MPCP protocol. Both rely on the EDSR protocol for wireless data communication, which extends the existing DSR implementation in the ns-2.

Broch et. al. [13] describe in detail the physical and the link layer model of the ns-2 also used in our simulations. We use the standard ns-2 settings and utilize WLAN 802.11 with a maximum data rate of 2 MBit/s as MAC. The Two-Ray-Ground propagation model has a maximum radio range of 250m.

All packets (both data and routing) sent by the routing layer are queued at the interface queue (IFQ) until the MAC layer can transmit them. The IFQ is a priority queue with drop tail characteristic. The IFQ queues routing packets at the beginning and data packets at the tail of the queue. Further on it can hold 15 packets at most and drops every additional incoming packet.

As shown in [13], ad hoc routing is most challenging with short pause times between consecutive node movements. Therefore, we set the pause time to zero to cause realistic node movements. Variable simulation parameters for different scenario files are the size of the simulation area, the number of nodes, their maximum speed, the size of the file download and the file replication rate. The number of nodes is varied between 50…200 nodes. To keep the node density constant, the simulation area varies between 1000x1000m^2 and 2000x2000m^2. In all scenarios every node has about 10 neighbors on average. In most cases this inhibits the network from being separated into several parts. Fully connected networks are important, as otherwise monitored events depend on the current network topology.

The nodes move according to the random waypoint mobility model. The maximum node velocities range from 0 m/s to 5 m/s. The size of the file download is either 2kB or 3MByte. The replication rate defines the fraction of nodes hosting the requested file. We generally use 10% as replication rate to have a sufficient number of possible sources even in small scenarios.

Each scenario is repeated 450 times with varying node movements and traffic scenarios. The error bars in the following graphs show the 95% confidence interval. All simulations contain only one initial file request and the following complete file download.

To be able to compare MPP and ORION we use the same simulation parameter set as described in [6] . The transmission range of nodes is reduced to 125m and the simulation area is 1000x1000m^2. If not otherwise stated, the number of nodes is 40 and the maximum node velocity is 2m/s. Simulations based on this scenario run at least 2500 times.

VI. EVALUATION AND ANALYSIS

A. MPP performance

We first evaluate the impact of the download procedure on the download success. As described in section IV, two different procedures exist, the static and the dynamic download. Figure 3 shows the download success against the number of nodes. To give insights, whether MPP is able to provide location based services (LBS) services, simulations contain requests to download service profiles. We assume the average size of a service profile to be about 2.0 kB of data [1]. Additionally we also performed simulations with object sizes of 3 MB, to study the suitability of MPP as a file sharing application and to be able to compare the performance of MPP and ORION.

Both the static as well as the dynamic procedure achieve 100% download success for short downloads (2 kB data). However the dynamic download algorithm shows an improved performance for the 3 MB download. Its success rate is beyond 90% for all network sizes while the static approach only achieves 60% success for larger network sizes.

Due to page restrictions, we do not show the download success for different maximum node velocities. Downloads of 2 kB data always have a success rate of 100%, independent from the maximum node velocities and the utilized download procedure. The dynamic download procedure is able to keep the success rate above 90% for velocities up to 20m/s. However the success rate for the static algorithm drops to 65% for 10m/s and achieves only insufficient 50% success rate for 20m/s. Simulations show that the static approach performs worse than the dynamic procedure for varying network sizes and for changing node velocities. Its success rate is up to 30% below the rate of the dynamic approach. The reason why the dynamic scheme outperforms the static scheme is, that with the dynamic scheme the requesting node can find nodes, which might have moved into its transmission range, while the node which was the nearest at the time of the first request moved out of it. Thus with the dynamic scheme the node is able to find the optimal download node at any time. In the case of the static scheme, the node resumes the download, after the primary downloading node fails, only from the second best. Concluding, we can say, that although more file requests are issued by one node in the case of the dynamic scheme, it causes less overhead, as it employs significantly shorter paths to download the requested content. Due to its better performance all following simulations only utilize the dynamic download procedure.

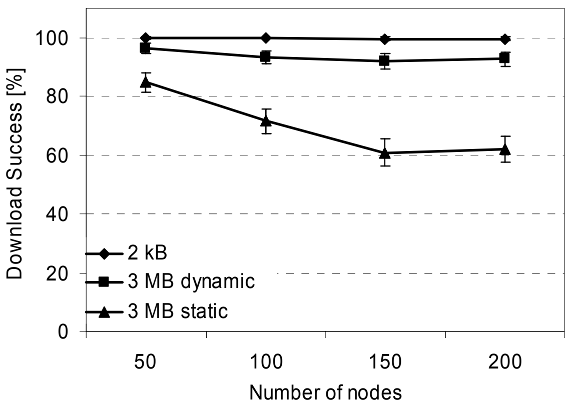

Figure 3: Comparison between static and dynamic download procedure against the number of node.

Based on the number of responses for each search request, we evaluate the performance of the search procedure of MPP. The fraction of responses means the total number of responses that the requesting node receives from the network over the total number of sharing nodes, which is 10% of the total number of nodes in this simulation.

To circumvent numerous ARP requests during simulations, scenarios contain an initial warm up phase before the actual performance measurements. Therewith the MPP search success reaches 80% of the theoretical limit.

There is still a deviation between the theoretical limit and the success rate. The source does not receive all FREPs or nodes do not receive the SREQ, because

- nodes replying with a FREP packet do not forward the initial SREQ and therewith following nodes do not receive the request
- packet collisions on the wireless medium occur
- return paths of FREPs show permanent errors, before the packets reach the source.

Simulations show, that the MPP search success is almost independent from the average node mobility. Although 100% of the sharing node can not be found, the requesting node can locate at least one sharing node in all of the simulation scenarios.

B. *Comparison with ORION*

The other important criterion of the search procedure in MANET is the search overhead that the search algorithm generates. To evaluate this characteristic, we compare MPP with the ORION approach. We performed simulations of MPP in the same simulation environment that was also employed by ORION.

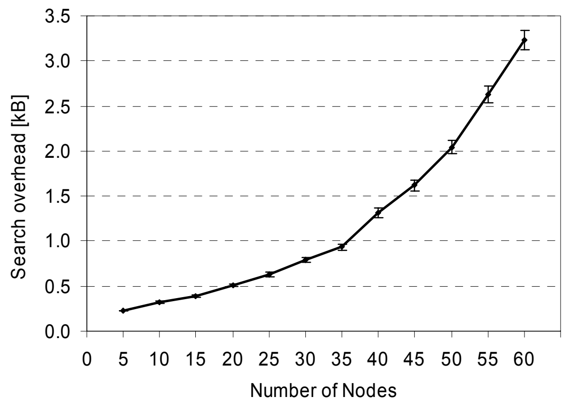

Figure 4: Search overhead against number of nodes.

The summarized size of all sending and forwarding EDSR packets at the network layer define the search overhead. As the search algorithm of MPP is integrated into the network layer, and the EDSR header contains all necessary data, we consider all these information as overhead. As MPP uses the standard MAC protocol, the search overhead does not include the MAC control overhead.

Figure 4 illustrates the dependence of the search overhead from the number of nodes in a scenario. The search overhead is given in kBytes and for individual requests. The number of necessary forwards increases linearly with the network size. Additionally the length of the source route contained within request packets increases linearly as well. Therewith the search overhead grows with $O(N^2)$ for increasing numbers of nodes and not linearly.

ORION transfers 27.5 Mbytes data for 100 search requests. 79% of that data contains payload traffic. That makes 59.1 kB search overhead for each request. We assume that the authors of ORION use their standard simulation parameters (see section V) for evaluation. Resulting, in our simulations, which are based on the same settings, MPP shows a better performance in terms of search overhead in comparison to ORION. With 40 nodes and 2 m/s maximum node velocity, the average generated traffic by one MPP search request is only 1.33 kB or 2,2% of the search overhead in ORION.

For the evaluation of the download success, the simulation parameters are equivalent to the setup for the search success evaluation. The results in Figure 5 show that MPP successfully downloads the service profile if the search is successful too. As the file size is relatively small, most of the file downloads require only one MPP file request to finish the download. However in case of primary path breaks during a download, additional FREPs or HREPs are necessary to reestablish connections to sharing nodes. Certainly this increases the download time, but it is always below one second. Therefore users never have to wait longer than one second for a response to their service request.

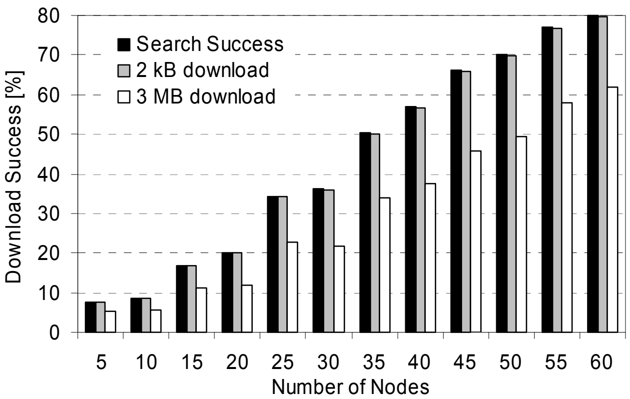

Figure 5: Search and download success against the network size.

As mentioned above, the average size of a MP3 music file is about 3 MB. To evaluate the file sharing capabilities of MPP and ORION, we run simulations with these file requests. As expected, the download success is about 20% below the search success. MPP must initiate several search requests before it finished a download. Thus the possibility of unsuccessful downloads increases. Interestingly, the gap between search and download success is the same for all network sizes greater than 30 nodes.

To compare the download success and the overhead between MPP and ORION, we refer to existing simulation results from [6]. ORION considers downloads as "failed" when the inquiring nodes runs out of alternative paths after a secondary request. MPP only initiates file downloads if there are responses to the original search request. Therefore we consider file downloads as "failed", if requesting nodes do not completely finish the MPP download procedure. No responses to MPP search requests count as "failed", because this is also counted as a "failed" request in the ORION evaluation.

Figure 6 shows the download success for varying maximum node velocities. The graph labeled MPP – 2 kB depicts the success for 2 kB downloads. It is an upper bound for the downloads, as all downloads are successfully finished.

For networks with low mobility, ORION performs better than MPP. However Figure 6 also shows that the download success of ORION for 3 MB files and 0.5m/s maximum node velocity is even higher than that of MPP with 2 Kbytes. A detailed analysis shows, that the poor results of MPP for low node mobility is because of low network connectivity in our simulations. Therewith a direct comparison between both algorithms for low node velocities is not appropriate.

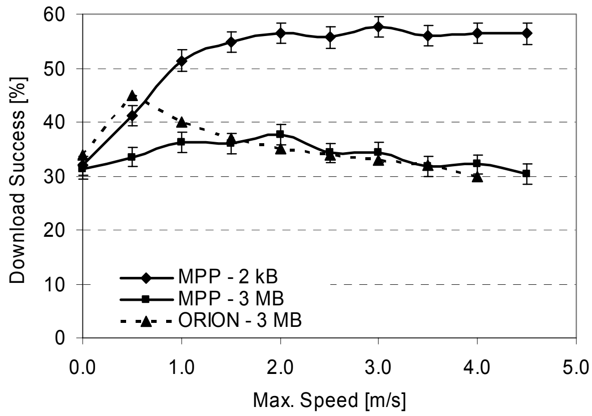

Figure 6: Download success against mobility.

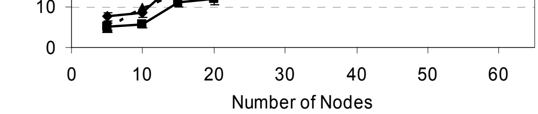

Figure 7: Download success against the network size.

Obviously both protocols achieve worse results for increasing node mobility. In any case MPP has at least a comparable performance as ORION. Therefore MPP as well as ORION provide a higher user satisfaction in comparison to of the shelf applications such as "Gnutella over Ad Hoc" [14].

As in the previous figure, the results for the 2 kB download in Figure 7 are again an upper bound for the download success of MPP. The download success raises with increasing numbers of nodes, as the network connectivity also increases. The problem of overloaded networks is negligible for network sizes below 60 nodes. Therefore simulations with less than 10 nodes mostly fail, because nodes are unable to connect to any neighbor. We can see that for a low number of nodes both behave equal, but for a network size above 35 nodes MPP performs better. MPP achieves a 80% success rate for the 2 kB download with 60 nodes, and the download of the MP3 file still shows a 60% success rate. However ORION only achieves a download success rate of below 50%.

The comparison of both algorithms leads to the conclusion that both algorithms perform equivalent for increasing node velocities. However MPP scales better than ORION for increasing number of nodes within a network. As realistic application scenarios are considered to employ more than 60 nodes, it could be concluded that MPP in general outperforms ORION.

The analysis of the ORION performance considers download overheads only for successful file transmission. This does not reflect the behavior of download procedures in MANETs correctly. The ORION protocol causes lots of unnecessary data transmissions, if it starts downloading a file but is unable to successfully complete it. Thus data transmissions for unfinished downloads have to be considered as overhead. In order to allow a fair comparison with ORION, Figure 8 and Figure 9 show two different graphs for the MPP download overhead. Graphs labeled “MPP” show the correct overhead, considering the unsuccessful file downloads as well. The other graph and the ORION graph only count the overhead of successful file downloads. In any case we can again observe the better scalability of MPP.

Both figures show the results for downloads of the 3 MB file, as this is the more difficult task. The 100% mark of the overhead depicts the relation, where the number of transmitted bytes is twice as large as the original download size of the file. Figure 8 and Figure 9 show the behavior of the download overhead against the number of nodes and the maximum node velocities. Considering only overhead of successful downloads, MPP and ORION show a comparable performance with respect to the number of nodes as well as maximum node velocities. For network sizes beyond 50 nodes and high velocities, MPP generates clearly less overhead than ORION.

For few nodes, connections are mostly between direct neighbors. Thus only packet and source route headers contribute to the overhead. This generates an overhead of about 40% for successful downloads. The download overhead raises linearly with the number of nodes, as possible sources are more far away from the requesting nodes.

Contrary to expectations, the download overhead decreases with increasing node mobility. Our analysis shows that the probability of closer sources increases with rising average velocities, while the possibility of connection breaks is still low. We can compute the average time, in which two nodes are within each others proximity, to $2\sqrt{2}R_0 / \pi v_{max}$. For $R_0 = 125m$ radio range and $v_{\max} = 4m/s$ node velocity, we can calculate the connectivity time to 28s, while the download of 3 MB requires only 15 s. Therewith the necessary number of connection reestablishments is minimal. Obviously, the download overhead will rise again with greater maximum velocities, as the connectivity time is decreasing linearly with $v_{\max}$.

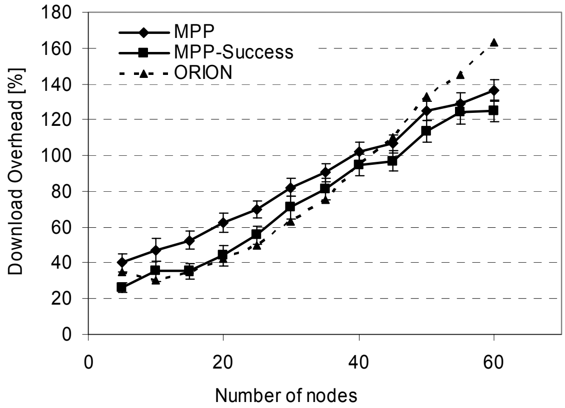

Figure 8: Download overhead against size.

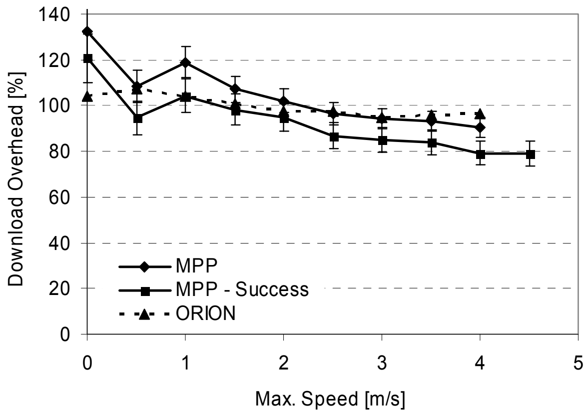

Figure 9: Download overhead against mobility.

As mentioned above, the download of 3 MB is occasionally unsuccessful. These downloads significantly contribute to the overall protocol overhead. Figure 8 and Figure 9 depict, that the overhead for successful downloads is always below the correct overhead, i.e. taking also unsuccessful transmission into account. Unsuccessful downloads occur even in small networks or networks without mobility. The extra overhead varies, but it adds always approximately 10% to 20% to the overhead of a successful download.

VII. CONCLUSION

In this paper we evaluated the performance of the MPP protocol suite. MPP is a novel solution for Peer-to-Peer communication over Mobile Ad Hoc Networks based on a tight coupling of the P2P overlay and the MANET routing protocol, in particular DSR. The proposed solution considers the dynamic physical topology of the network nodes and allows context-based routing by extending the routing goal to other objects than IP addresses such as files.

To prove the benefits of the performance of the MPP approach, we performed several simulations based on ns-2.

Thus we are able to compare MPP with the only known similar solution ORION.

MPP only requires 2.2% of the search overhead of ORION. For increasing network sizes, MPP outperforms ORION in terms of search success and download overhead. Simulations also show, that MPP achieves 100% success for service profile downloads, if networks have a sufficient connectivity.

Overall the MPP protocol stack provides the fundamentals to realize P2P applications in a mobile ad-hoc network. Compared to other approaches it allows to build powerful P2P applications with minimum data overhead. Therefore MPP performs well and can provide Internet-like applications. Furthermore, by integrating routing of both networks, the system can easily deal with link failures and fast moving peers.

So far MPP is relying on DSR as a routing protocol in MANETs. We intend to generalize our approach for different routing protocols in the future.

References

[1] R. Schollmeier, I. Gruber, F. Niethammer. Protcol for Peer-to-Peer Networking in Mobile Environments. Proceedings of the ICCCN 2003. Dallas, TX. 2003

[2] C. Perkins, E. Royer. Ad hoc On demand Distance Vector Routing. Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999.

[3] D. Johnson, D. Maltz, D. Dynamic source routing in ad hoc wireless networks, in Mobile Computing (ed. T. Imielinski and H. Korth), Kluwer Academic Publishers, pp 153-181, The Netherlands, 1996.

[4] M. Papadopouli, H. Schulzrinne. Effects of Power Conservation, Wireless Coverage and Cooperation on Data Dissemination among Mobile Devices. Proceedings of the ACM Symposium on Mobile Ad Hoc networking (MOBIHOC 2001). Long beach, CA. 2001.

[5] S. Lim, W. Lee, G. Cao, C. R. Das. A Novel Caching Scheme for Internet based Mobile Ad Hoc Networks. Proceedings of the ICCCN 2003. Dallas, TX. 2003

[6] A. Klemm, C. Lindemann, O. Waldhorst. A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks, Proc. IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall), Orlando, FL. 2003.

[7] K. Hildrum, J. Kubiatowicz, 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.

[8] 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.

[9] 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.

[10] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, 1999.

[11] K. Fall, K. Varadhan, editors. ns notes and documentation. The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997.

[12] The CMU Monarch Project. The CMU Monarch Projects Wireless and Mobility Extension to ns. Work in Progress, December 2003.

[13] J. Broch, D. Maltz, et. al. A Performace Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. Proc. of ACM MobiCom 1998, Dallas, USA, October 1998.

[14] A. Oram. Peer-To-Peer, O'Reilly Press, 2001.

[15] R. Schollmeier, A. Dumanois. Peer-to-Peer Traffic Characteristics. EUNICE 2003. Budapest, Hungary. Sepetember 2003.