Peer-to-Peer Overlay in Mobile Ad-hoc Networks

Marcel C. Castro[1], Andreas J. Kassler[1] Carla-Fabiana Chiasserini[2], Claudio Casetti[2], and Ibrahim Korpeoglu[3]

Abstract Wireless multi-hop networks such as mobile ad-hoc (MANET) or wireless mesh networks (WMN) have attracted big research efforts during the last years as they have huge potential in several areas such as military communications, fast infrastructure replacement during emergency operations, extension of hotspots or as an alternative communication system. Due to various reasons, such as characteristics of wireless links, multi-hop forwarding operation, and mobility of nodes, performance of traditional peer-to-peer applications is rather low in such networks. In this book chapter, we provide a comprehensive and in-depth survey on recent research on various approaches to provide peer-to-peer services in wireless multi-hop networks. The causes and problems for low performance of traditional approaches are discussed. Various representative alternative approaches to couple interactions between the peer-to-peer overlay and the network layer are examined and compared. Some open questions are discussed to stimulate further research in this area.

Department of Computer Science, Karlstads University, Universitetsgatan 2, SE-651 88, Karlstad, Sweden {Marcel.Cavalcanti, Andreas.Kassler}@kau.se ·
Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy {chiasserini, casetti}@polito.it ·
Department of Computer Engineering, Bilkent University, 06800 Ankara, Turkey korpe@cs.bilkent.edu.tr.

1 Introduction

A mobile ad-hoc network (MANET) is a collection of autonomous mobile nodes that communicate using wireless links without support from any pre-existing infrastructure network. In such a multi-hop network, nodes operate as both end hosts and routers, forwarding packets wirelessly towards other mobile nodes that may not be within the direct transmission range of each other. MANETs are formed with the key motivation that users can benefit from collaborations with each other. Wireless mesh networks (WMN) are comprised of a wireless mesh backbone formed of quasi-stationary wireless mesh routers which wirelessly relay packets generated by (mobile) mesh clients, that connect to the wireless mesh routers like to normal access points. WMNs are emerging as an attractive infrastructure for next generation wireless access networks and they share many properties with MANETs such as multi-hop forwarding. While MANETs typically operate standalone and more autonomous, Internet access for MANETs and WMNs is desirable. Multi-hop networks such as MANETs or WMNs have been considered to support future ubiquitous and pervasive computing scenarios, and therefore will be intrinsic part of the future Internet.

Recently, applications based on the Peer-to-Peer (P2P) communication paradigm are increasing in popularity. Examples are popular file-sharing applications (e.g., Kazaa [40], Gnutella [55]), upcoming P2PSIP solutions for Voice over IP, or P2P video streaming that use P2P techniques to form an overlay on top of existing networks. P2P computing refers to technology that enables two or more peers to collaborate spontaneously in a network of equals (peers) by using appropriate information and communication systems without the necessity for central coordination. In that sense, P2P networks are overlay networks typically operated on infrastructured (wired) networks, such as the Internet. However, the P2P overlay network is dynamic, where peers come and go (i.e., leave and join the group) for sharing files and data through direct exchange. Such peer-to-peer communication paradigm will be very important in wireless multi-hop networks as centralized servers might not be available or located in the Internet. Therefore, P2P will be an interesting alternative for decentralizing services or making its own local resources available in the multi-hop network to serve local user communities.

P2P overlay networks in the Internet and mobile ad-hoc networks share many key characteristics such as self-organization and decentralization due to the common nature of their distributed components [32]. They also share a high degree of dynamicity as nodes can join and leave the network at any given time. These common characteristics lead to further similarities between the two types of networks: both have a frequently changing topology caused by nodes joining and leaving dynamically. Also in a MANET terminals are mobile and communication follows a hop-by-hop connection establishment.

The common characteristics shared by P2P overlays and MANETs also dictate that both networks are faced with the same fundamental challenge, that is, to provide connectivity in a decentralized and dynamic environment. Thus, there exists a synergy between these two types of networks in terms of the design goals and principles

of their routing protocols and applications built on top: both P2P and MANET routing protocols and applications have to deal with dynamic network topologies due to membership changes or mobility.

In addition, P2P overlays over the Internet rely on the IP routing infrastructure, which is resource rich especially in terms of bandwidth availability. Mobile ad-hoc networks, instead, are rather limited in bandwidth, and a high maintenance traffic, as it is used currently in structured overlay networks, will lead to scalability problems when legacy P2P services are used ”as-is” in multi-hop environments. Thus, one of the main issues is how to efficiently provide the same kind of P2P services implemented in legacy wired networks in multi-hop networks, and how to enable efficient overlay services and applications on the resource constrained wireless multi-hop networks.

The common characteristics, challenges, and design goals between P2P overlays and mobile ad-hoc networks point to new research directions in wireless networking, that is, to exploit the synergies between P2P overlays and multi-hop networks such as MANETs. There are several examples where knowledge on interactions between P2P and MANET can either help to realize more efficient P2P networks and services on top of multi-hop networks or will lead to the design of better and more scalable routing protocols [70, 52, 8, 24]. Understanding such interactions will also help to clarify, what support from routing layer shall be required for scalable operation of P2P on top of heterogeneous mobile networks.

The remaining part of this chapter is then organized as follows. In Section 2, we give a brief overview on structured and unstructured overlay networks. We introduce wireless multi-hop networks and highlights key properties of wireless operation and multi-hop forwarding. The challenges encountered while deploying P2P services in mobile ad-hoc networks are detailed in Section 3. Section 4 provides a detailed survey of related approaches including work on both unstructured (e.g., flooding based protocols, unstructured key lookup, and proactive search routing) and structured (e.g., topology dependent and topology independent) P2P overlays for MANETs. Recent studies, such as ORION [38], MPP [26], P2PSI [30], ZP2P [37], VRR [8], SSR [24], CrossROAD [18], MADPastry [70], MeshChord [7], and Hashline [60] will be introduced. Additionally, the respective advantages and disadvantages are evaluated. Section 5 introduces important P2P application scenarios for MANETs, such as decentralized name service (e.g., MAPNaS [71] and P2PNS [2]), overlay-based multicast (e.g., XScribe [19]), and multimedia services (e.g., P2PSIP [20]). Finally, Section 6 concludes the chapter.

2 Overview on Peer-to-Peer and Ad-Hoc Networks

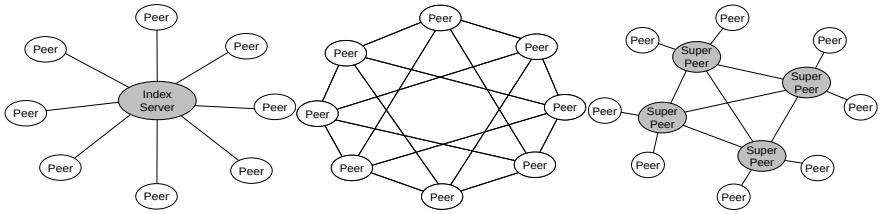
Wireless multi-hop networks feature several peculiar aspects which significantly differentiate them from other wireless systems and pose serious technical challenges. In this section, we highlight the main characteristics of these systems and discuss some of their most challenging issues, i.e., wireless multi-hop communication, mobility, and traffic routing in multi-hop networks.

2.1 Peer-to-Peer Overlay Networks

We begin however with a brief overview on peer-to-peer networks. There are numerous peer-to-peer overlay networks proposed with very different architectures and protocols. The architectures for P2P overlays can be categorized into two main classes: unstructured P2P overlays and structured P2P overlays.

Unstructured overlays do not impose a rigid relation between the overlay topology and where resources or their indices are stored. This has a number of advantages like; easy implementation and simplicity, supporting dynamic environments and keyword search (instead of exact match queries). But the major drawback of such overlay is scalability problem. Search operation for a resource may take a long time and consume network resources extensively, since most of the time there is no relation between the name of resources and their locations. Depending on the degree of centralization, unstructured P2P overlays are usually classified into three sub-categories: 1) hybrid decentralized overlays such as Napster, Publius, and Bittorent [44, 51] (Figure 1a); 2) purely decentralized overlays such as initial version of Gnutella and Free Haven [55, 56] (Figure 1b); and 3) partially centralized overlays such as Gnutella version 0.6, Fasttrack/Kazaa, Morpheus, Overnet/eDonkey2000 [55, 40, 29] (Figure 1c). In all categories, the resources (or services) are totally distributed to peers and there is usually no relation between the locations of resources and the network topology. But depending on the category, central or distributed indices, clustering, super-peer concept, caching and replication can be used [42, 1].

A common feature provided by peer-to-peer overlay networks is a lookup service (i.e., searching for resources) handling flat identifiers with an ordinary query-response semantic. Such a service is often implemented using DHTs (Distributed Hash Tables), such as CAN, Chord, Pastry, and Bamboo [53, 61, 57, 54] . Unlike unstructured P2P networks with their random topology, DHTs impose a structure on the overlay topology by no longer choosing routing table entries arbitrarily. Instead, routing table entries have to satisfy certain criteria depending on the respective DHTs. At the core of each DHT lies the ability to route a packet based on a *key*, towards the node in the network that is currently responsible for the packet's *key*. This process is referred to as *indirect* or *key-based* routing. This structure enables DHTs to introduce an upper bound on the number of overlay hops towards the node currently responsible for the packet's key. This upper bound is commonly $O(logn)$, with



(a) Hybrid decentralized (b) Purely decentralized (c) Partially centralized

Fig. 1: Unstructured P2P overlays

n being the number of nodes in the network. This bound is achieved through routing strategies employed by the respective DHTs. Those strategies include reducing the Euclidean distance in the overlay ID space to the destination in each overlay routing step (e.g., CAN [53]), halving the numerical distance to the destination in each routing step (e.g., Chord [61]), or increasing the length of the matching prefix/suffix between the current node's overlay ID and the key in each overlay routing step (e.g., Pastry [57] and Bamboo [54]). Although DHTs can route packets very efficiently in comparison to unstructured P2P networks, they usually induce higher overhead due to the need for maintenance traffic of their routing tables. The maintenance traffic routine can be initiated by network change, such as in Chord and Pastry, or within certain periodicity regardless of network status, such as in Bamboo. While reactions to changes in the routing layer operate on very small timescale, reactions to changes in overlay structure are not so fast. In [54], the approach to use periodic updates has been shown to be beneficial during churn or in dynamic network, since it does not cause management traffic bursts during congestion. As we will show in Section 4, management traffic can impact network performance when applied to bandwidth limited wireless environments. However, as argued by [25], DHT approaches outperform unstructured approaches when the number of nodes, the number of objects, or the query rate increases, since they do not introduce flooding in the network.

2.2 Characteristics of Wireless Multi-hop and Mobility

Wireless multi-hop communication has many use cases, both in standalone deployments, but also to extend the reach of infrastructure, e.g. hotspots. Such wireless communication involving potentially multiple intermediate nodes poses several fundamental challenges, also stemming from hidden and exposed terminals resulting in packet loss, and high and variable delay and thus low performance in general. Several of these factors play a significant role in any wireless communication scenario. However, as communication is extended to multiple hops, several new wireless issues come into play. Single hop communication results in most cases in a single collision and interference domain. In contrast, in multi-hop cases the roles of col-

lision and interference become more complex and depend on many factors such as radio environment, modulation schemes, transmission power, or sensing ranges. As a result, adjacent links and even links further separated, affect each other during transmission and they might have to share the wireless channel. In single channel networks, a two-hop configuration hence effectively halves the available bandwidth. Other links still within interference range also might affect links further down a multi-hop path, reducing the link bandwidth even further. Such behavior has many subtle performance implications to higher layers such as TCP [31], which are not visible in single hop networks.

To alleviate such problem, in WMNs mesh routers may be equipped with multiple radios (such as of-the-shelf 802.11a/b/g cards) to simultaneously transmit/receive over different orthogonal frequency channels. However, to fully exploit the available resources, it is necessary to develop mechanisms to effectively assign available channels to a limited number of radio interfaces per node. If a mesh is rather unplanned or channel allocation is done poorly, interference might be quite high leading to the same problems.

Another problem area is mobility of nodes, quite common to MANET scenarios. as a result, the network might become disconnected for a long period or the high mobility might lead to frequently changing communication paths. Such effects impose several challenges such as long delays, disrupted communications, and intermittent connectivity to communication protocols. As a result, most higher layer protocols such as TCP cease functioning or show dramatically low performance. Therefore, commonly assumed communication design principles such as the permanent availability of a dedicated end-to-end path have to be reconsidered leading to new communication paradigms that are significantly more delay tolerant than common approaches such as digital postal service through *store-carry-forward* message delivery. This style of delivery carries information between intermittent communication opportunities, and might be an attractive alternative of enabling communication where it is otherwise impossible. Such communication paradigms might also be useful for other contexts such as satellites networked into an inter-planetary Internet [6] or postal service like data delivery into rural areas where communication infrastructure is not available [13]. Instead of assuming an always on connection, communication entities rather carry information between intermittent communication opportunities, leading to the opportunistic communication paradigm.

2.3 Traffic Routing in Multi-hop Networks

Routing is an essential function for Internet and also very important for wireless multi-hop networks, e.g. MANETs. Indeed, while at the MAC and physical layer it is commonly assumed that the IEEE 802.11 standard is adopted, a large number of different proposals on traffic routing have been presented within the IETF (The Internet Engineering Task Force) and are still under discussion.

Typically routing protocols in MANETs can be classified in flat and hierarchical schemes. Flat routing protocols distribute information as needed to any network node that can be reached or receive information. No effort is made to organize the network or its traffic, only to discover the best route hop-by-hop to a destination by any path. Hierarchical routing protocols, instead, group nodes together by function into a hierarchy, e.g., if there are powerful nodes, they may be selected as backbone routers, while lower powered node may be used for access purposes.

In the context of wireless ad-hoc and mesh networks, flat routing schemes have been far more successful than hierarchical solutions, thus, below, we focus on flat routing and review the most relevant schemes that have been proposed in the literature as well as those solutions that are mostly used in practical implementations. On the other hand for more opportunistic communication style in delay tolerant networks, new type of more probabilistic routing protocols have been developed as the main challenge is to cope with long periods of disconnection and opportunistically exploit communication possibilities.

2.3.1 Topology-based Schemes

The routing protocols falling in this category exploit information related to the network topology. They can be further classified in (i) reactive protocols and (ii) proactive protocols. Reactive schemes create routes only when required by a source node. Once a route is established, it is maintained by a route maintenance procedure until either the source does not need the route any longer or there is no available path in the network. Examples of reactive solutions are the well known Ad-hoc On Demand Distance Vector (AODV) [49] routing and Dynamic Source Routing (DSR) [33] protocols. In AODV, when a route to a new destination is needed, the node broadcasts a RREQ (Route REQuest) message to find a route to the destination. A route can be determined when the RREQ reaches either the destination itself, or an intermediate node with a ”fresh enough” route to the destination. A ”fresh enough” route is a valid route entry for the destination whose associated sequence number is at least as great as that contained in the RREQ. The route is made available by unicasting a RREP (Route REPly) back to the origination of the RREQ. Each node receiving the request caches a route back to the originator of the request, so that the RREP can be unicast from the destination along a path to that originator, or likewise from any intermediate node that is able to satisfy the request. While AODV builds and maintains routing tables at every node, DSR obtains and encodes the source route in each packet header to the destination. It follows that DSR leads to a greater overhead with respect to AODV, although it can handle both unidirectional and bidirectional links and allows nodes to store more than one route for each source-destination pair.

Proactive schemes, instead, attempt to continuously maintain consistent, up-to-date routing information from each node to any other node in the network. As in AODV, every node has one or more tables, which are used to store routing information; upon topology changes, a node propagates update messages throughout the network in order to maintain a consistent view. Hence, in highly dynamic net-

works the overhead of proactive approaches is significantly higher than with reactive schemes, however when proactive solutions are applied, nodes always store routes to any possible destination in the network. Among the most interesting proactive solutions, there are the Optimized Link State Routing Protocol (OLSR) [14] and BATMAN (Better Approach to Mobile Ad-hoc Networking) [46], which deserve special attention because, along with AODV, are the protocols typically used in practical implementation of MANETs and mesh networks.

OLSR is a link-state routing protocol which exploits Hello and Topology Control (TC) messages to discover and then discriminate link state information throughout the ad-hoc network. Individual nodes use this topology information to compute next hop destinations for all nodes in the network using shortest hop forwarding paths. More specifically, using Hello messages the OLSR protocol performs a distributed election of a set of multipoint distribution relays (MPRs), such that there exists a path to each of its 2-hop neighbors via a node selected as an MPR. These MPR nodes then source and forward TC messages which contain the MPR selectors. Such approach has several benefits: the forwarding path for TC messages is not shared among all nodes but varies depending on the source, only MPRs source TC messages, and not all links of a node are advertised but only those which represent MPR selections.

BATMAN has been specifically designed for wireless mesh networks. The basic idea is to divide the knowledge about the best end-to-end paths between nodes in the mesh to all participating nodes. Each node perceives and maintains only the information about the best next hop towards all other nodes. Thereby, the need for a global knowledge about local topology changes becomes unnecessary. Additionally, an event-based but flooding mechanism prevents the occurrence of contradicting topology information and limits the amount of topology messages flooding the mesh (thus minimizing overhead of control-traffic). Since it adopts a hop-by-hop forwarding approach, BATMAN may be particularly suitable for networks whose connectivity level is not very high.

2.3.2 Geographic-based Routing

Geographic routing protocols do not require knowledge of the network topology but rely on geographic position information, i.e., each node must be able to determine its own location and the source has to be aware of the location of the destination [62]. With this information, a message can be routed to the destination following different approaches. Greedy Perimeter Stateless Routing (GPSR) [36] tries to bring the message closer to the destination at each step, using only local information (greedy forwarding). Additionally, in regions of the network where such a greedy path does not exist, GPSR recovers by forwarding in perimeter mode. That is, a packet traverses successively closer faces of a planar subgraph of the full radio network connectivity graph until reaching a node closer to the destination, where greedy forwarding resumes. Alternatively, one can consider another notion of progress toward the destination, namely the minimum angle between neighbor and destination, as in Com-

pass Routing [39] which however is not loop free. Clealy, whenever the destination node is mobile, geographic routing may be highly inefficient and the exchange of nodes location may lead to an exceedingly high overhead.

2.3.3 Probabilistic Routing

This approach has low complexity and is particularly suitable for networks with spotty connectivity, i.e., the so-called opportunistic networks. The basic idea is that context information, such as the users work address, the probability of physically meeting with other users or visiting particular places, can be exploited to identify suitable forwarders based on context information about the destination. Here, the mobility of nodes is exploited to deliver information from one node to another when they come into mutual communication range. Examples of protocols falling in this category are the Probabilistic ROuting Protocol (PROPHET) [41] and MaxProp [5]. PROPHET is an evolution of the epidemic approach that introduces the concept of delivery predictability. The delivery predictability is the probability for a node to encounter a certain destination. The delivery predictability for a destination increases when the node meets the destination, and decreases (according to an ageing function) between meetings. Transitivity is also taken into account, i.e., if node X frequently meets node Y, and node Y often meets node Z, then nodes X and Z have high delivery predictability with respect to each other. Also, when two nodes X and Y meet, they exchange their delivery predictability to destinations of the messages they store in their buffers, and messages are transfered from, say, X to Y only if Y's delivery predictability is higher than the one of X. The same technique is used by MaxProp, which, in addition, exploits information about frequently visited places.

3 Challenges of Deploying P2P Services in Mobile Ad-hoc Networks

The suitability of MANETs for applications that rely on a P2P architecture for information exchange presents designers with several challenges. Indeed, not only do mobile nodes require content delivery but they also act as content providers. Mobile users are expected to offer data services in an effective manner, despite the scarcity of bandwidth and the intermittent connectivity due to the highly-dynamic nature of MANETs. Below, we list some of the technical challenges in delivering information to mobile users depending on a P2P organization.

Bandwidth Constraints

The challenge of introducing P2P concepts in multi-hop networks is that P2P overlays designed for the wired Internet rely on the IP routing infrastructure, which is resource rich especially in terms of bandwidth availability. As we have seen in Section 2.2, mobile ad-hoc networks are however rather limited in bandwidth. Therefore, a high maintenance traffic, e.g. as it is used currently in structured overlay networks, will lead to scalability problems when legacy P2P services are used ”as-is” in multi-hop environments. One of the main issues is therefore how to efficiently provide the same kind of P2P services implemented in legacy wired networks in multi-hop networks, and how to enable efficient overlay services and applications on the resource constrained wireless environment. As it is presented in Section 4, several approaches try to overcome such challenge by integrating, or applying cross-layering techniques between the P2P and the MANET routing layer.

P2P Overlay Maintenance

Keeping the overlay routing table of each node up to date is one of the main tasks of a DHT system. Efficient routing depends on routing information being current and consistent. Invalid entries cause unnecessary overhead because of misrouted messages and suboptimal routing. To avoid these inconsistencies, DHT protocols employ maintenance mechanisms to keep the routing tables up to date. Typically, nodes probe their neighboring nodes via periodic ping request and response messages to learn whether they are still available or not. In MANETs, such maintenance traffic further contributes to congestion and collisions. As nodes mobility might lead to topology changes in the MANET routing layer, there might be potential for misrouted messages if the overlay routing and the MANET routing have inconsistent topology information. Also, triggering such maintenance traffic during network rerouting further contributes to network instability. To this end, cross-layer and integrated approaches are applied by, for example, exploiting the network routing messages (such as CrossROAD [18]) or cache information (such as SSR [24]) in order to maintain the P2P overlay.

Network Resiliency

In P2P networks with structured overlay, DHTs are considered to be very resistant against node failures. Backup and recovery mechanisms, that use distributed redundant information, ensure that no information is lost if a node suddenly fails. Depending on the subjacent DHT topology, the DHT experiences a reduced routing performance until the recovery has finished.

When DHT protocols are used in an ad-hoc environment, resilience is as a very important issue. The resilience of a DHT determines how much time may pass before expensive recovery mechanisms have to be evoked. As the quality of connections in ad-hoc networks is highly dependent on the environment and on the nodes mobility, nodes may often become temporarily inaccessible. If the recovery process is started too early, an avoidable overhead is caused if the node becomes accessible again. However, if the topological structure allows the DHT protocol to delay recovery mechanisms without losing routing capability, these costly recovery measures can be avoided and the maintenance costs of a DHT can be significantly reduced. As an example described in Section 4.3.1, [12] studies a compromise made between overlay management traffic in the overlay and network congestion to find a balance between lookup efficiency and management traffic overhead.

The worst case scenario is represented by a network where long delays and disrupted communications exist, as mentioned in Section 2.2. In this case, a node which is partly available and unavailable over a longer period of time can stress the whole network because of numerous join and leave procedures. Note that this scenario can easily be provoked by node movement along the network perimeter and, clearly, resilience mechanisms are needed to counteract the negative effects of this condition.

Routing Stretch

Unlike the P2P overlay in the Internet, where the neighbor is directly reachable using an underlying routing protocol, in the P2P overlay in MANETs scenario, contacting the neighbor may require going through multiple (wireless) hops. For this purpose, a pointer is maintained for every overlay's neighbor as a path through the network, consisting of a set of physical links from the node hosting the pointer to its overlay's neighbor.

When routing to a destination via DHTs, the node resorts to simple greedy routing: it selects the overlay's neighbor that makes the most progress in the ID space, and then forwards the packet along the pointer. Forwarding along this pointer can be achieved either through a source route inserted by the sender (e.g. SSR [24]) or through embedded state in the network in the form of incremental source routes to the overlay neighbor (e.g. VRR [8]). Both techniques will be discussed later. When the packet reaches the overlay-neighbor, it repeats the same greedy routing process until the packet makes it all the way to the destination. Therefore, routing proceeds at two levels: along the overlay from one overlay neighbor to another, and then from one overlay neighbor to another along the pointer source route via hop-by-hop through MANET routing protocols.

The ratio between the cost of selected route using the overlay-neighbor to the optimal shortest path routing through the MANET is defined as the *routing stretch* metric. Small routing stretch means that the selected route is efficient compared

to the shortest path route. This is a key quantitative measure of route quality used by the P2P overlay, and affects global resource consumption, delay, and reliability. Thus, minimizing routing stretch is a critical issue for a multi-hop environment as both delay and packet loss increase significantly with the growth of the number of hops in the physical path.

Exploiting Heterogeneity

Another important point while deploying P2P overlay is which nodes should participate in the overlay given that not all nodes in a network may be overlay members [73]. While typically nodes in an overlay are initially placed manually, nodes may also dynamically and automatically decide to join and make services available. This issue may be especially important in multi-hop environments because overlay participation may be dictated by topological location which might change over time. Note, that other (e.g., physical) constraints may drive the decision to participate in the overlay. For example, nodes with limited power may not wish to act as overlay routers for other nodes.

Query Propagation

The propagation of query messages in the network is a critical aspect of the information sharing mechanism in P2P networks. Indeed, there are two contrasting requirements that arise in MANETs. On the one hand, queries for information must be forwarded by relays until they reach nodes holding such information, and some redundancy in forwarding is necessary to compensate for the unreliable nature of broadcast transmission of queries (i.e., no acknowledgments). On the other hand, congestion deriving from excessive spreading of queries and reply duplication must be limited. The simplest solution for query propagation is, of course, plain flooding of requests, but this is hardly viable in tightly-meshed, bandwidth-hungry wireless networks where congestion is more than likely. More refined approaches, are among others:

1. *Limiting query range*. The introduction of a query time to live (TTL) can shorten the reach of broadcast queries. A balance should be stricken between small values of TTL, which limit the success probability of a query, and query load.
2. *Smart relaying*. By forcing each relay to wait for a query lag time before rebroadcasting the query, the propagation of a request can be halted if a node in the neighborhood returns a response in the meantime (thus making any further query propagation useless). Coupling the query lag time with a smart selection of intermediate nodes for query rebroadcast may turn out to be very beneficial. As shown in [45], the Preferred Group Broadcasting (PGB) limits the network load through local, receiver-based decisions to rebroadcast a message. Intermediate nodes still wait for a lag time before rebroadcasting, however its length depends on the value of the signal-to-noise ratio (SNR) associated to the received message.
3. *Target selection*. Steering the queries toward the right direction is, of course, the main remedy against broadcast storms. Targeting a specific node that is known to store the information can be exploited at the application level, by leveraging the

knowledge of the address and position of the last node encountered, which happened to cache the desired information. However, node targeting proves very inefficient in a MANET built by rapidly-moving nodes and running fast-dynamics applications. For this reason, a better approach is targeting *areas* of the network where the requested information is more likely to be cached, as proposed in [23].

Cooperative Content Caching

In purely decentralized overlays, a highly debated issue addresses the most appropriate caching strategy in an environment where a cache-all-you-see approach is clearly unfeasible but where the availability of sought-after information from nearby nodes is the key to success. This issue can be addressed through distributed caching strategies where nodes may cache highly popular contents that pass by, or record the data path and use it to redirect future requests [69]. Another viable solution is to eliminate information replicas among neighboring nodes [27], which however may require the nodes composing the MANET to coordinate their caching decisions. An interesting aspect is also how to minimize data access cost when network nodes have limited storage capacity. The scheme proposed in [63] makes use of cache tables that, in mobile networks, need to be maintained in a similar vein as routing tables.

As is clear from the above discussion, solutions to cooperative caching in mobile multi-hop networks, which are distributed and rely on lightweigth communication protocols, are still to be found. Finally, when different copies of the same information are injected in the network, maintaining cache consistency among the different nodes becomes a critical issue [28, 9].

Information Distribution and Survival

A final, critical issue pertains to achieving a desired distribution of the information within an area: regardless of how the information is distributed at the outset, the system should be able to identify where the information should be stored in the network area. In addition, a node storing the information acts as provider for that information; of course, this role may exact a high toll from nodal resources in terms of bandwidth or power consumption; it is advisable that the role of content provider be handed over to neighboring nodes quite frequently, without altering the information distribution. One or more nodes running out of power may affect the distribution of information and disrupt the P2P structure. Therefore, regardless of the initial information distribution, and of the density of nodes, information should never be allowed to die out. Related to the information survival is the evaluation of the minimum number of copies of a specific information that can satisfy users' needs (i.e., in terms of information retrieval time or response rate).

Security

Deploying security mechanisms in P2P networks is quite difficult due to the characteristics of P2P paradigm such as anonymity, decentralization, self-organization and frequent disconnections. Security in P2P over mobile ad-hoc networks is even more challenging due to node mobility and easy access to wireless channels. Most security solutions require use of public keys for authentication, shared secret estab-

lishment, or integrity checking, and hence somehow depend on a public key infrastructure (PKI) [35].

PKI is needed by asymmetric cryptography to establish the validity of the public keys. For this purpose, PKI stores digital certificates that attach a public key to the name of its owner by the digital signature of a trusted third party called the Certification Authority (CA). The management of certificates is a complex duty that requests a substantial infrastructure, especially in large-scale applications. Integration of PKI and CAs, or a similar security infrastructure, into P2P over MANET is a challenging task due to ad-hoc and infrastructureless nature of the network and lack of centralized entities. Even in P2P networks with servers (hybrid centralized or partially centralized - see Section 2.1), these servers usually do not fully control the peer behaviors as much as servers can do in a conventional client-server model. Thus, the centralized architecture of PKI may introduce several important problems that contradict with the important characteristics of the P2P networks and MANETs. Additionally, PKI and security services may introduce substantial amount of control traffic into the network, which means more load to bandwidth-limited wireless channels of MANETs.

4 Overview of P2P Solutions for Mobile Ad-hoc Networks

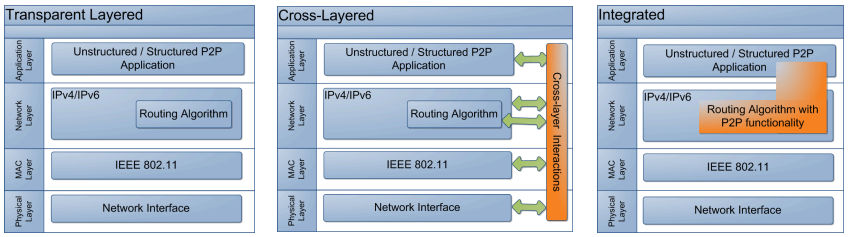
In the following, we present and discuss various approaches to improve performance of peer-to-peer communication in wireless multi-hop networks, such as MANETs or WMNs. As several proposals try to integrate different layers to reduce bad interactions, we will first give an overview on different principles that guide the various integration and interaction possibilities, both in the area of unstructured (in Section 4.2) and structured (in Section 4.3) though there may be some overlapping similarities between the two.

4.1 Integration Principles between P2P and MANET Routing Layer

One of the main differences between P2P and MANET is related to the level where they operate: P2P is essentially focused on building and maintaining overlay network connections at the application level, while the main focus of MANET is to provide multi-hop connectivity among wireless mobile nodes at the network level [58]. Due to the characteristics of multi-hop communication and the low resource availability in such networks, simply deploying a P2P overlay protocol as is on top of MANET routing layer (as shown in Figure 2a) might cause poor performance, significant message overhead and redundancy in communication. The performance penalties of such transparent layering are better detailed in Section 4.3.1, where a packet level performance analysis of Bamboo over static multi-hop networks has been conducted.

One alternative for avoiding bad interactions between those layers is the paradigm of cross-layer design, as shown in Figure 2b. Here, information from, for instance, the routing or MAC layer is made available at the peer-to-peer layer or vice versa in order to improve the performance. Various approaches implement different cross-layer interactions, as detailed in Sections 4.2 and 4.3. As a result, a cross-layered design could offer a significant performance improvement if compared to the simple layered approach.

Another alternative to increase performance is to integrate peer-to-peer layer with routing layer beyond the strict layering rule [15], as shown in Figure 2c. Typically new routing mechanisms (such as key-based routing) are developed, and try to implement peer-to-peer concepts in the routing layer itself. In the next sections we provide an overview of these approaches, by also trying to evaluate the key features of each of them.



(a) Layered design (b) Cross-layered design (c) Integrated design

Fig. 2: Design choices of P2P and MANET integration

4.2 Unstructured P2P Networks for MANETs

Actually, several works on the convergence of peer-to-peer systems and mobile ad-hoc networks have dealt with the straightforward implementation of unstructured P2P overlays in MANETs. Those approaches combine ad-hoc routing and unstructured overlay flooding, usually using the route discovery mechanisms of the ad-hoc routing protocol to locate the desired resource in the network.

One of the first documented system is 7DS [47], which attempts to enable P2P resource sharing and information dissemination in mobile environments, been rather a P2P architecture proposal than a practical application.

In [38], ORION aims at providing peer-to-peer services in a MANET, bringing a general purpose distributed lookup service and enhancing file transmission schemes to enable file sharing in MANETs. ORION applies the integration (Figure 2c) of Gnutella-style [55], flooding into the AODV [48] ad-hoc routing to locate requested files in the network. With ORION, each node in the MANET has a local repository containing the files that the node is sharing. When a node wants to locate a certain file, it issues a query message that is broadcasted through the network. Whenever a node receives such a query message, it sets up the reverse route to the originator just as AODV does with its route request (RREQ) packets and retransmits the query message to its physical neighbors. Furthermore, each intermediate node checks its local repository for any files that match the description (e.g., file name, key words, etc.) specified in the query message. If such files are found, the node will send a response message containing the identifiers of all matching files back to the requester using the AODV-style reverse route. Each intermediate node on the response path will also update its file information cache with the file identifiers contained in the response message and the provider (i.e., the sender of the response message). After the requester has received a response, it will then send a data request for the desired files to (one of) the provider(s) using the AODV-style routes discovered during the search. The provider will then divide the requested file into blocks and send data packets containing the various blocks of the requested file back to the requester. The basis of ORION is AODV, and it concentrates only on file sharing applications, providing an application layer routing protocol which causes unnecessary overhead.

The MPP (Mobile Peer-to-Peer) protocol [26] is also proposed as a file sharing system in MANETs. In contrast to ORION, MPP adapts the overlay structure to the physical MANET structure via a cross-layer communication channel (Figure 2b) between the MANET network layer and the P2P layer. The MPP protocol stack reuses existing network protocols as much as possible. For node-to-node communication, the protocol utilizes an enhanced version of the Dynamic Source Routing (DSR) protocol [59]. More specifically, EDSR (Enhanced Dynamic Source Routing) combines Gnutella-style flooding and DSR ad-hoc routing. For the transportation of user data it uses HTTP over TCP. To connect the application layer protocol with the network layer protocol (EDSR), the Mobile Peer Control Protocol (MPCP) is used. The MPCP is the inter-layer communication channel between the application and the network layer. Using the 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 to the corresponding protocol all incoming and outgoing requests and responses, except the file exchange itself. Besides file sharing applications, MPP also intends to provide location aware services.

In MPP, when a node wants to locate a desired file, it will issue a search request that is flooded throughout the MANET, leveraging the EDSR route discovery process. Whenever a node receives such a search request, it will communicate with its application using the MPP protocol stack to see if the application can provide a matching file. Each intermediate node adds its own node address to the search request to create a DSR-style route and retransmits the search request to its physical neighbors. If the application can provide the requested file, a reply message will be send back to the requester using the reverse path information as contained in the search request. After the requester has received a reply, it will download the desired file from the provider using HTTP. Responses to queries performed by MPP's nodes (and also ORION's nodes), result in a network-wide broadcast of search requests, giving a routing algorithm complexity of $O(n)$ [21], where n is the number of nodes. This is clearly a downside of both approaches as they might not scale to both growing network sizes and increasing request rates.

Hoh et al. proposes in [30] a P2P file sharing system over MANETs based on swarm intelligence, called P2PSI. Basically, it is an hybrid push-and-pull system composed by two processes. In the advertisement process (push), each hotspot [1] periodically advertises a *seed* message containing digest information about files to be shared within a limited area (e.g. as determined by the hop count). Every node can independently make the decision on when to advertise and which files to advertise to its neighbors, and such decision can be based on e.g. a ranking system to maximize the number of report delivered [67]. In order to reduce seed message size, Bloom filter technique [3] is applied as a method for summarizing the list of shared files. Upon a node receives a *seed* message, it will cache this information. When been queried, the node that has the cache of the file information will send a reply to the

[1] In [30], authors consider a quite large portion of peers to be free-riders, who only retrieve files from others without making contributions to share files. Therefore nodes willing to share files are called hotspots and they are assumed to provide almost all popular files and some private collections.

querying node. In the discovery process (pull), the node willing to search for a file, first checks if it has cached the desired file information. If not, the node deploy query messages, forwarded at intermediate nodes based on their pheromone table, to find the identity of the node holding the desired file. The pheromone table records the pheromone intensity on each neighbor link, which denotes the probability of routing a query message via that neighbor based on the number of hops traversed by reply message.

According to [30], the search accuracy of a cross-layered approach, such as P2PSI, is always higher than that of a layered one, as request success ratio decreases at larger network sizes due to increased overhead for the layered approach. In order to avoid such redundancy overhead between P2PSI file discovery and network route discovery process, a cross-layered design (Figure 2b) is used integrating P2PSI and ARA (Ant-based routing) protocol [4]. The advantage of such design was experimentally observed by implementing P2PSI in the ns-2 simulator and comparing it against two cross-layered design service discovery protocols: *CL_dsr* and *CL_dsdv* [64]. The results show that as the network size and node mobility increase, the request success ratio of the P2PSI outperforms *CL_dsr* and *CL_dsdv*. Indeed the performance of request success ratio of *CL_dsr* deteriorates as it utilizes flooding to search for a file which becomes the performance bottleneck when the network size grows. The same behavior emerges in *CL_dsdv* since it fails to converge as the node mobility increases.

In order to reduce the heavy overhead of always broadcasting search requests in the MANET, zone-based protocols, such as ZP2P (Zone-based P2P by [37]) have been proposed. ZP2P is based on the concept of local zones, determined by a fixed hop-count. When a node is interested in a certain object, it will first check its local cache to see whether any of its zone members can provide the desired object. However, in case the requested object is not available in the node's own zone, it will initiate a *bordercast* of the request through its border nodes, i.e., to those of its zone members that are exactly k hops away. In case a border node finds that there are no members in its zone that could provide the requested object, it will continue the *bordercast* by forwarding the request to its own border nodes. This process continues until either a predefined TTL expires or the whole network has been searched.

By introducing the concept of local zones into the P2P search process, some of the network-wide broadcasts may become unnecessary. However, whether or not a requested file can be provided by nodes inside the requester's own zone depends entirely upon chance. Especially in larger networks, the cases where a request could be satisfied locally can be expected to be rare [70]. Hence, the utility of local zones will evidently not scale with growing network sizes. The propagation of requests using *bordercasts* can lower the overall traffic as a certain number of inner nodes might not have to forward the requests. Nonetheless, with growing network sizes, the *bordercast* process will quickly encounter the same problems of a regular broadcast as the number of zones that need to be contacted also increases. Furthermore, the efficacy of a *bordercast* depends entirely on factors such as the zone radius and the node density inside the zones. In networks with low or medium node density, it is likely that the routes from the center node of a zone to its border nodes will

involve most (if not all) of the inner nodes. Thus, in such networks, the *bordercast* will closely resemble a regular broadcast, and the performance of ZP2P can be expected to be worse than that of a regular broadcast, due to the additional continuous (update) advertisement messages that need to be exchanged. Although not explicitly addressed in [37], nodes need to periodically re-issue their advertisements to take into consideration the effects of node mobility on zone memberships. This will cause ZP2P to generate additional traffic, with respect to a regular broadcast application.

4.3 Structured P2P Networks for MANETs

The concept of DHT was first proposed by Plaxton [50] without the aim to address P2P routing problems. But, it soon proved to be a useful substrate for large distributed systems and a number of projects have been proposed to build Internet-scale facilities leveraging the DHT concept. On the other hand, ad-hoc networks gained great importance due to the increasing occurrence of scenarios which do not have a centralized infrastructure. Whenever there is a need for a scalable data management without any infrastructure, the combination of ad-hoc network and DHT technology seems to be a promising solution [32]. The questions, whether this is beneficial, and how current solutions perform such combination will be discussed in the following sections.

4.3.1 Transparent Layered DHT on Top of Broadcast Based Ad-hoc Routing Protocol

Deploying a DHT directly on top of an existing broadcast based ad-hoc routing protocol does not require any changes to the routing or overlay layer. In that approach, every file name and peer is hashed to a key by standard hash algorithms (e.g. SHA-1 [22]). Every peer should maintain a small routing table of size $O(logn)$, in which each entry directs to an intermediate peer closer to the requested key. The peer closest to the requested key knows the address of the actual peer storing the requested file. In order to route to these intermediate peers, standard MANET routing protocols are deployed which usually acquire topology information using broadcast, increasing the routing algorithm complexity to $O(nlogn)$. As described by [21], this is due to the fact that network routing protocols in MANET introduce complexity of $O(n)$ to find the route between every two peers, although there are only $O(logn)$ peers needed in the P2P overlay.

In order to maintain the correctness of each overlay routing table, peers need to periodically communicate with each other through overlay management protocols. These protocols should be triggered more frequently in MANETs due to mobility and characteristics of the underlying physical networks. Otherwise, routing information at the overlay might not be consistent. In [12], the performance of Bamboo is

evaluated in a static multi-hop environment common to ad-hoc networks. When deploying Bamboo over MANET following a layered approach (Figure 2a), the overlay network forms a virtual network in the application layer while the underlying network is transparently managed by MANET routing protocols such as AODV.

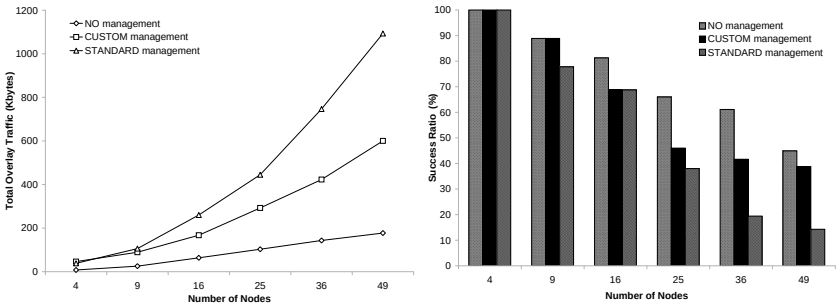
Bamboo uses proactive management traffic in order to maintain the network structure. Neighbor ping is generated by every node in order to make sure that the node can still reach its one-hop neighbors in the overlay, and it is also used to maintain a RTT estimation for retransmission timeout calculations. Nodes also perform leafset update by periodically choosing a random node from its leafset, and execute a leafset push followed by a leafset pull. Bamboo considers that two nodes share the same level when one node contains the other node in its routing table. Therefore, the local routing table update is used to exchange the node information in that level. Data storage updates are also performed in order to maintain the desired number of replicas among the peers.

However it is expected that the proactive management maintenance introduced by Bamboo increases network traffic, and consequently as the network grows, high congestion will be experienced. In order to find a balance between management traffic in the overlay and network congestion, three different configurations for Bamboo management traffic were compared in [12]; 'no' management, 'standard' management (used by [54]), and 'custom' management. Table 1 presents the parameters used by each configuration. The comparison carried tries to find a balance between lookup efficiency and management traffic overhead. Too frequent management traffic will lead to high overhead in multi-hop environments and thus lead to network congestion. No management, on the other hand, will leads to low lookup efficiency.

Simulations were performed using ns-2 over different scenarios, where the nodes were positioned on a grid at a distance of 200m, with 250m of transmission range and 500m of carrier sense range using two ray ground as radio propagation model. The transmission rate is set to 11Mbps, and the basic rate to 1Mbps. The AODV-UU routing protocol was adopted using default settings proposed by [65]. Simulations were performed for 60 seconds without bootstrapping period. During the experiments, every 2 seconds, each node generates a 500-byte PUT message with a random key to store data in the overlay. All nodes also try to acquire random selected keys that are located on other nodes generating a 32-byte GET message every 2 seconds.

Table 1: Bamboo Management Timers (secs)

	NO	**Standard**	**Custom**
Leafset Update	-	1	5
Local Routing Update	-	5	10
Global routing update	-	10	20
Data Storage update	-	2	6
Neighbor Ping	0,5	0,5	0,5



(a) Management Overhead (b) Success Rate

Fig. 3: Impact of Bamboo Management traffic

Figure 3a presents the total Bamboo management overhead, which represents the aggregation of the overlay management traffic including: neighbor ping, leafset update, routing table update, and data storage update, for the three different scenarios. As expected, the overhead introduced by Bamboo increases with number of nodes, and is much higher for 'standard' timeout settings compared to 'no', and 'custom' management. This is mainly due to the aggressiveness of periodic updates required by Bamboo to monitor the status of other nodes in the overlay and update the overlay data structures. On the other hand, in the case of 'no' management, each node does not generate periodic updates, but neighbor ping is still performed in order to maintain the leafset peers.

Figure 3b also illustrate the success rate behavior of Bamboo over the scenarios chosen by [12]. As the number of nodes increases, network load increases and success ratio decreases accordingly as illustrated in Figure 3b. For example, in the 36 nodes grid, the success ratio is 61%, 41% and 19%, respectively for 'no', 'custom' and 'standard' management. The lower success ratio for higher number of nodes can be explained by the higher percentage of management and routing overhead in order to maintain the overlay structure, as shown in Figure 3a. The ability to find the destination nodes which are responsible for the specific keys degrades as management overhead increases network contention. This results in higher number of resent and dropped packets over the wireless links due to network congestion and consequently problems in the routing layer, as shown in more details over the simulation results presented in [12].

Other related publications, such as [17] which deploys Chord over MANET routing protocols, also indicate that simply deploying a standard MANET routing layer does not scale with increasing number of clients, network size, and mobility. The reasons are manifold such as the characteristics of multi-hop communication, the consistency problem between the two routing layers, and the design assumptions for MANET routing protocols which assume traffic characteristics unlike those of structured overlay protocols.

4.3.2 Integrating DHT over the Network Layer

As illustrated in Section 4.3.1, the characteristics of the underlying ad-hoc network protocol has great effect on the performance of the overlay as the DHT induces a constant flow of control and query messages. An optimized interaction between ad-hoc network and DHT is essential to create an efficient combination. There are several approaches proposed in the literature that try to exploit similarities between ad-hoc network and DHT in order to integrate them in a system with higher performance, by also reducing the overheads. The examined approaches analyzed here are VRR [8], SSR [24], CrossROAD [18], MADPastry [70], MeshChord [7], and Hashline [60].

Virtual Ring Routing (VRR) [8], proposed by Microsoft Research Centre, is a networking routing protocol which pushes peer-to-peer concepts to the network layer itself. Caesar et al. argue that VRR brings benefits when implemented over MANETs, as it balances the load of managing hash-table keys across nodes, and avoids flooding of routing messages through the network. Based on Pastry [57], VRR organizes the nodes into a virtual ring ordered by their identifiers. Each node maintains a small number of routing paths to its neighbors in the ring.

In VRR, node identifier are fixed, unique and location independent. To maintain the integrity of the virtual ring with node and link failures, each node maintains a virtual set (vset) of cardinality r (predecessor and successor nodes). The routing path between a node and each of its virtual neighbors is called vset-path. The routing table also maintains the physical neighbor set (pset) with the identifiers of the nodes that it can directly communicate with at the link layer. Such information is gathered through broadcast of hello messages periodically. The routing information for a vset-path is also stored on the nodes along the paths. Then, a node maintains a routing table with information about the vset-paths to its virtual neighbors, other vset-paths that are routed through the node, and the pset of physical neighbors. As described in [8], VRR requires $rp + k$ routing table entries per node on the average, where p is the average path length, and k is the number of physical neighbors. Since node identifiers are random [2] and location independent the virtual neighbors of a node will be randomly distributed across the physical network. So, the probability that a random node has a path to a random destination is $O(rp/n)$. Therefore, a packet is expected to reach a node that has a vset-path to the destination after visiting $O(n/(rp))$ nodes.

Unlike routing protocols that forward packets based on destination address, VRR nodes route packets to destination identifiers (keys) by forwarding them to the next hop towards the path endpoints whose identifier is numerically closest to the destination identifier from among all the endpoints in their routing table. An advantage of such scheme is that these keys can identify application objects instead of just VRR nodes. Control messages to set up new vset-paths are routed using existing vset-paths avoiding the flooding on the network.

[2] VRR hashes the node current IP address in order to obtain the node identifiers

Scalable Source Routing protocol (SSR) [24] brings the same concept of VRR while trying to integrate the P2P overlay into the network layer. But while VRR does not assume any specific MANET routing protocol integration, SRR combines the Dynamic Source Routing protocol (DSR) [34] in the physical network with Chord routing in the virtual ring formed by the address space. Fuhrmann states that SSR trades off shortest path for a reduced amount of state information, leading to less maintenance overhead. Therefore, besides the successor, SSR's nodes store the addresses of $O(logn)$ additional nodes at exponentially spaced distances to reduce the average request path length from $O(n)$ to $O(logn)$, where n is the number of nodes in the network.

Following the DSR concept, data packets of SSR contain a source address, a destination address and a source route. However according to SSR design, the source route does not have to span the entire path from the source to destination. When the virtual ring has been established, SSR can route messages to any destination. By constructing the route cache, each node contains source routes to the node's neighbors in the virtual ring. Beside that, the caches will contain source routes to other destinations also. For example, all nodes that are part of a source route in the cache can be viewed as potential destinations. When routing a packet, the respective node chooses the (intermediate) destination from its cache that is physically closest to itself and virtually closest to the final destination of the packet. It appends the source route from its cache to the packet's header. The nodes along this source route can then forward the packet using the source route in the packet. This routing step is repeated at all intermediate nodes and all subsequent destinations until the packet has reached its final destination. If the virtual ring has been formed consistently, this routing algorithm is guaranteed to succeed for any source and destination pair.

To maintain the virtual ring consistency in SSR, all nodes must have valid source routes to their respective virtual neighbors; e.g. its predecessor and successor in the address ring. The nodes need also to have information about their physical neighborhood, information which is gathered through a periodic beacon message (e.g. hello message). The state maintenance of the virtual ring continues until all nodes have mutually correct virtual neighbors, in order to guarantee network convergence. In order to reduce the routing stretch, SSR's nodes use the source routes in their routing caches to prune unnecessarily long source routes, e.g. routes contain cycles or a shorter sub-path to one of the nodes in the source route is known (short cut). However, as discussed by [70] the effectiveness of this source route pruning entirely depends upon the available cache entries and there are no guarantees as to how well the source routes in the system can be pruned.

CrossROAD is proposed by [18] as a way to reduce communication overhead introduced by Pastry when deployed over mobile ad-hoc networks. Different from VRR and SSR integrated approaches, a cross-layered architecture defining interactions between P2P and routing layers allows CrossROAD to exploit additional information to optimize the overlay management. These interactions are handled by the Network Status module (NeSt) [16], an external data sharing module, which provides interfaces for cross-layer interactions throughout the protocol stack. Each node running CrossROAD piggybacks advertisements of its presence in the overlay

into routing messages periodically sent by OLSR. Thus, each node in the network becomes aware of the other peers in the overlay network. Then, each node in the overlay maintains a routing table of size $O(n)$. Since each node knows all nodes taking part in the overlay, the sender of a specified message can directly identify the closest destination for the selected key, and subsequently use the OLSR protocol at the network layer to deliver the message through the shortest path ($O(1)$ virtual hops in the overlay).

[18] states that such mechanism reduces the overhead required to build and maintain DHTs in legacy systems such as Pastry, however at the cost of additional overhead in the OLSR layer. However, no remote connections are required by CrossROAD to initialize the overlay routing table, neither in case of disconnection events or network partitioning. It directly exploits the network routing protocol that collects topology changes periodically sending its LSU (Link State Update) packets, and directly updates its own routing table and the related abstraction in the NeSt. In this way CrossROAD becomes aware of topology changes with the same delays of the routing protocol. Nevertheless, it is worth to mention the lack of results regarding scalability of CrossROAD to both growing network sizes and node mobility.

In order to take physical location into consideration, MADPastry is proposed by [70]. MADPastry integrates (Figure 2c) the application layer Pastry and the reactive ad-hoc routing protocol AODV. The concept of random landmark [66] is used to create physical clusters where nodes share a common overlay ID prefix. Since there are generally no stationary nodes available in MANETs, MADPastry works without any fixed landmark nodes. Instead, it uses a set of landmark keys, which are simply overlay IDs that divide the overlay ID space into equal-sized segments. Nodes associate themselves with the temporary landmark node that is currently closest to them (e.g. as determined by the hop count) by adopting its overlay ID prefix. For that purpose, temporary landmark nodes send out beacons periodically. These beacons are broadcast and whenever a node overhears a landmark beacon, it stores the current landmark node's ID and the distance to it as given by the hop count of the beacon. As broadcast imposes serious network burden, landmark beacons are only propagated within the landmark's own cluster, i.e. beacons are only forwarded by nodes belonging to that cluster.

When a MADPastry's node intends to advertise a resource, it will now insert the resource descriptor under two different keys. The first key is the regular hash key (of the resource's URI, etc.) inserted into the network. To obtain the second key under which the resource descriptor is stored, the regular resource key is altered to make sure the descriptor will be stored in the resource host's own MADPastry cluster. For this purpose, the resource key's prefix is replaced with the host's own cluster prefix (current landmark node's ID). Hence, intra-cluster communication can be expected to travel only short physical paths, as lookups process will try to find the corresponding resource descriptor in its physical vicinity (local cluster members). However such optimization might be useful for popular files or standard services that are hosted by multiple nodes. Only if this local lookup provides no (appropriate) answer, will the request be forwarded as in a regular network-wide lookup. Following this process, the first key remains fixed during the lifetime of a node,

while the second one can change depending on the node's position in the physical network.

To be able to route packets along the network, MADPastry nodes maintain three different routing tables: a standard AODV routing table for physical routes from a node to specific target nodes, as well as a *stripped down* Pastry routing table and a standard leafset for indirect routing. Differently from Pastry routing table which consist of $log_{2^b}N$ rows, the *stripped down* Pastry routing table only needs to contain $log_{2^b}K$ rows, with K being the number of landmark keys. Using such approach, MADPastry avoids the expensive Pastry routing table maintenance overhead, but it deliberately sacrifices the $O(logn)$ bound on the number of overlay hops during a key lookup. MADPastry also perform a proactive routing table maintenance, by periodic pinging its 'left' and 'right' leaf. According to Zahn, this is necessary to guarantee overlay routing convergence. The remaining routing entries are gained by overhearing data packets. Then, the accuracy of the Pastry routing tables and leafsets largely depend on the number of packets that a MADPastry node receives or overhears. With the idea of proximity awareness using random landmarking, physical clusters of nodes sharing a common overlay ID prefix are created, avoiding longer overlay hops per lookup.

MeshChord, proposed by [7], is an specialization of Chord applied to wireless mesh networks, where the availability of a wireless infrastructure, and the 1-hop broadcast nature of wireless communication are taken into account while performing key lookup. In MeshChord, routers are assumed to be stationary, but they can be switched on/off during network lifetime. If a client in the mesh network wants to find a certain resource, it sends a key lookup message to its reference mesh router (a mesh router within its transmission range). The reference router forwards the resource request in the DHT overlay according to the rules specified by the Chord protocol, until the resource query can be answered. As in Chord, in a n-node system, each MeshChord's node maintains information about only $O(logn)$ other nodes, and resolves lookups via $O(logn)$ messages to other nodes.

MeshChord explores location awareness by assigning IDs to peers according to their coordinates, accomplished by, for example, the use of GPS receivers. Besides that, MeshChord also takes advantages of 1-hop broadcast communication by overhearing lookup request packets in order to speed up lookup operation. Then, by overhearing a lookup request at the MAC layer, a node can reply to it if the requested ID is comprised between its ID and the ID of its predecessor in the unit ring.

It is worth observing in [7] that location awareness tends to decrease the lookup operations under dynamic network conditions. In fact, location-aware ID assignment tends to rule out the possibility of having close-by peers in the physical network which are far-away in the overlay (e.g. in Chord, possibly corresponding to the last fingers in the finger table). However, MeshChord achieves a considerable reduction in message overhead, and improvement in query response time while utilizing location awareness and overhearing strategies.

Hashline [60], a DHT-based file sharing system for wireless ad-hoc networks, also integrates the P2P query functionality with the network routing. Hashline is

able to answer location queries and also discover and maintain routing information that is used to transfer files from a source peer to another peer. In this way, it enables the proposed P2P file sharing system to run on an ad-hoc collection of wireless nodes without requiring a separate MANET routing protocol at the network layer.

The basic idea in Hashline is the adaptation of the CAN [53] P2P routing protocol. Unlike CAN, however, [60] uses a one-dimensional space, called *hashline*, into which keys and node IDs are mapped. The hashline is divided hierarchically into segments so that each node is responsible from one segment. The values (location information) of the keys falling into a segment are stored in the corresponding node responsible for that segment. The relationship between segments can be considered as a tree consisting of parents and children, so that the hashline segment of a parent spans the hashline segments of all its children.

In [60], when a node would like to find the location of a file with key k, the node forwards the query to one of its children if k falls into the hashline segment of one of the children. Otherwise the query is forwarded to the parent. Hence a tree based routing is used. At the end, the node that is responsible for the hashline segment including the key receives the query. That node knows the location of the file and also the route to that location. It answers the query together with the location and route information. The requester can then download the file from that location using the learned route. Hence the download operation does not require a different routing protocol to find the route to the location where the file is stored. In this way, queries and downloaded files are efficiently routed in the network. However, the operations performed to keep the tree-based routing state up-to-date when a node leaves or joins are quite costly. Hence the proposed protocol is suitable for low mobility wireless networks. As described by [60], the number of routing table entries mantained by each Hashline's node is at most k, where k is the number of of physical neighbors.

4.4 Summary and Comparison of the Solutions

As seen, a number of different approaches exist that could potentially be used as building blocks for large scale distributed network applications in multi-hop networks, such as MANETs or Mesh Networks. The varying characteristics of the presented approaches sometimes make it difficult to compare them directly against each other. Therefore, Table 2 intends to assess the different approaches according to:

- Fusion with Underlay: integration principle between P2P and MANET interactions;
- P2P overlay protocol: inspired P2P protocol;
- Routing Algorithm: routing algorithm deployed at the network layer;
- Overlay Adaptation: overlay topology reaction to network change;
- Periodic management: periodic management information exchanged among peer nodes at the overlay layer ;
- Location Awareness: use of location information to construct the overlay;

- Proposed Applicability: proposed applicability and use cases considered;
- Prototype Implementation: prototype implementation availability.

It is interesting to analyze that all unstructured approaches utilize a Gnutella-like protocol. Structured approaches are mainly based on Chord and Pastry (as Bamboo is inspired by Pastry). Regarding routing algorithms, most approaches studied rely on reactive routing protocols, such as AODV, DSR, and ARA. Proactive routing algorithms, such as CrossROAD, appear to be very expensive in terms of resource usage and routing table maintenance traffic injected into the network.

The cross-layered or integrated design (Figures 2b and 2c) of unstructured P2P overlays and ad-hoc routing is an intuitive and simple solution for the discovery of objects in MANETs. It is a straightforward approach as the changes and enhancements to the underlying ad-hoc routing protocols are minimal since, for example, reactive MANET routing protocols already have the capability of broadcasting requests and directly replying to the requester. However, first and foremost, the obvious disadvantage of such approaches is their poor scalability when network size grows. The main reason is that network-wide broadcast of search requests scales to neither growing network sizes nor increasing request rates. P2PSI and ZP2P try to scale to large MANETs under mobility by applying ant colony behavior and zone-based broadcasting, respectively.

Despite Bamboo/AODV, the DHT-based protocols avoid duplicated overhead through integration or cross-layering design. They also try to avoid broadcasting whenever possible, and optimize their DHT entries by overhearing packets. A significant difference among these systems is the use of location aware information by MADPastry and MeshChord, compared to the other DHT-based protocols. MADPastry exploits the concept of random landmarking to create overlay clusters, while MeshChord assumes that nodes are stationary, have their own position information available, and uses MAC layer overhearing to reduce search latency. Furthermore, since reply and file transfer messages are unicasted for all unstructured and structured approaches, their reliability depends entirely on the scalability and performance of the chosen (reactive or proactive) ad-hoc routing protocol.

Table 2: Assessment of related approaches

Solutions	Fusion with Underlay	P2P Overlay Protocol	Routing Algorithm	Overlay Adaptation	Periodic Management	Location Awareness	Proposed Applicability	Prototype Implementation
ORION	Integrated	Gnutella-like	AODV	N/A	N/A (Not Applied)	No	File sharing	No
MPP	Cross-layered	Gnutella-like	DSR	N/A	N/A	No	File sharing and location aware services	No
P2PSI	Cross-layered	Gnutella-like	ARA	N/A	N/A	No	File sharing	No
ZP2P	Cross-layered	Gnutella-like	AODV	N/A	N/A	Y (via local zones)	File sharing	No (SDL specification)
Bamboo/AODV	Layered	Bamboo	AODV	Proactive	Leafset and routing table	No	File sharing, decentralized name service, P2PSIP	Yes (Linux based)
VRR	Integrated	Pastry	AODV	Reactive	Leafset table	No	File sharing, decentralized name service, P2PSIP	Yes (Windows based)
SSR	Integrated	Chord	DSR	Reactive	Successor and predecessor	No	File sharing, decentralized name service, P2PSIP	Yes (Linux based)
CrossROAD	Cross-layered	Pastry	OSLR	Proactive	OLSR topology information	No	Multicast-based, white board applications	No
MADPastry	Integrated	Pastry	AODV	Reactive	Leafset table	Y (via cluster formation)	File sharing, decentralized name service	No
MeshChord	Cross-layered	Chord	DSR	Reactive	Predecessor and finger table	Y (via coordination information)	File/Resource sharing	No
Hashline	Integrated	CAN	Custom	Reactive	N/A	Y (via tree formation)	File/Resource sharing	No

5 P2P Application Scenarios for Mobile Ad-hoc Networks

The P2P solutions presented in the previous Section provide ways to deploy efficient distributed resources in MANETs using flooding, or key-based routing. These solutions are important building blocks to realize P2P applications in MANETs. In this section, we detail their use in important applications and services such as decentralized name service, overlay-based multicast, and multimedia services.

5.1 Decentralized Name Service

Nearly all Internet applications use persistent, human-readable names for users, hosts, and services. In the current Internet, this is done using the the Domain Name System (DNS), which is a centralized, distributed system with a single root of trust.

In peer-to-peer systems such as P2PSIP [20], it is useful to have human-readable, user-friendly names, but a centralized naming service is an undesirable choke point. It is difficult to implement a centralized service in a MANET, therefore it is interesting to decentralize service using P2P concepts.

As an example, MAPNaS, a decentralized name service for MANETs, is proposed by [71] in order to identify a resource (e.g. a file, a service, etc) by a unique resource key that is mapped into the logical DHT space. Due to the lack of a fixed network topology, there are no dedicated resource directory servers. Instead, every node functions both as a resource host (e.g. of its own files and services) and as a resource directory for certain remote resources.

While mobile devices often have limited hardware and maybe storage capabilities, the design goal of MAPNaS is to keep the architecture simple, where nodes store the resource descriptors (the resource key along with the specific network address of the resource) they are responsible for in their local MAPNaS repository. Furthermore, every node advertises which resources it is willing to share through MAPNaS. When a node in the network wants to make a local resource (e.g. a service, a file, etc.) available to other nodes in the network, it assigns a hash key to that resource, e.g. by hashing the resource's URI. Using that key, the node will then construct a resource descriptor consisting of the resource key and the physical network address (e.g. IP address) of the resource provider (in this case, the node address). Using the DHT, the descriptor is routed to the node currently responsible for the resource key. That recipient node will then store the resource descriptor in its local repository.

Resource discovery with MAPNaS works similarly to the resource advertisement process. First a lookup request is sent to the node currently responsible for the hash key of the resource's identifier. Then, the eventual destination node will check its local repository and send back the matching resource descriptor (or multiple descriptors in case several nodes are hosting the same resource). As the DHT in MAPNaS is realized through MADPastry [72], location replications of resource descriptors are restricted to MADPastry's clusters.

In traditional SIP networks the main task of a SIP server is to resolve an Address of Record (AoR) to the current IP address (Contact URI) of a user. This name resolution usually depends on Domain Name Server (DNS). P2PNS [2] presents a distributed name service using DHT to resolve AoRs to Contact URIs without relying on DNS and central SIP servers. Apart from this decentralized name resolution the call setup is based on the standard SIP protocol. In P2PNS there is a separation between the overlay layer (key-based routing), the data storage layer (distributed hash table), the name resolution layer (P2PNS Cache) and the protocols, that utilize the name service (like SIP or DNS). Hence, the specification of the key-based routing protocol is independent of P2PNS, and key-based routing solutions discussed earlier could be easily applied in the MANET environment.

The P2PNS architecture comprises a name resolution and caching layer (P2PNS Cache) on top of an overlay which provides key-based routing and DHT services. In P2PNS, a two-stage name resolution mechanism is proposed to efficiently handle frequent IP address changes. A user chooses an arbitrary name as AoR (e.g. name@p2pname.org). Then a mapping from the selected AoR to the corresponding nodeID [3] is stored in the DHT. In this case the name resolution layer first queries the DHT for the nodeID (given the user's AoR) of the destination node and in a second step resolves this nodeID to the current IP address of nodes.

5.2 Overlay-based Multicast

Overlay-based multicast is one option to implement multicast at the P2P layer. Usually, multicast protocols are classified as operating at the network layer, like routing protocols, or at the application layer, where 'application' denotes all possible layers above the transport. Overlay-based multicast runs only at nodes involved in the related application, and it just requires standard unicast support from the routing level. There are basically two approaches: 1) structured approach and 2) unstructured approach. In the structured approach, a multicast routing structure, like a tree, is established at the overlay level. Hence parent-child relationships are defined between peers making up the tree and the packets are forwarded over these peers towards the receiver peers which are also part of the overlay tree. In the unstructured approach, no such structure is established and used. Instead the sender has to know which receivers are interested in the packets and sends them to each receiver using a different mechanism, such as unicasting the same packet to each receiver. This requires the sender to know the potential receivers of the multicast data, which can be achieved through a multicast group membership protocol.

Applying existing P2P multicasting solutions developed for wired and infrastructure-based networks to MANETs will not work efficiently due to various reasons dis-

[3] Every peer chooses once a 160 bits nodeID for joining the overlay. This nodeID is retained even if the peer changes its IP address or leaves the overlay from time to time. The DHT allows to resolve the nodeID to the current IP address of a peer.

cussed before. Therefore, existing solutions must be adapted or new solutions must be developed.

XScribe [19] is an structured P2P multicasting protocol for ad-hoc networks. It is based on the well known P2P multicasting protocol Scribe [11], which was developed for wired P2P networks. XScribe can be used to implement various multicasting services and works together with CrossROAD in order to obtain the network topology.

In XScribe, the sender is required to know receivers of a multicast group using a membership management protocol. The sender obtains this knowledge using a cross-layer approach, where each multicast receiver sends its bitmask (indicating which groups it is interested in) embedded into the CrossROAD routing packets. When the sender has a packet to sent to a group/topic (hence to the receivers that are interested in that topic), the sender directly sends the packet to each receiver using the CrossROAD DHT overlay. Therefore, the packet is unicasted to each receiver without the need to setup a tree or any other multicasting structure before sending data.

Even though this seems to be inefficient, simulation results in ad-hoc networks show that XScribe performs better compared to deploying the original Scribe protocol over MANETs with standard routing due to the reduced routing stretch between peer nodes in the overlay structure.

5.3 Multimedia Services

In P2P file sharing applications, the main concern is to locate files to a given query. Once located, the user can decide to download the file, which then is downloaded out-of-band (i.e., not through the P2P overlay itself, but through the underlying networking and transport mechanisms). Hence for file and resource sharing P2P applications, data transport is not the main concern.

For P2P multimedia services, however, the situation is different. For non-realtime media, the media is typically located, downloaded and then played back from the local disk, in contrast media streaming provides faster response time at less client storage. Media streaming, however, requires a different type service provisioning and transport from the underlying network. Certain amount of network resource such as high bandwidth and controlled delay. To guaranteed smooth delivery admission control [10] needs to be implemented in order to provide real time streaming, which requires also tight control and end-to-end delay.

Providing P2P media services over ad-hoc networks is challenging due to the characteristics of multi-hop forwarding and the wireless medium (see Section 2.2. On the one hand, if some peers become hot-spots as media uploaders, the upload capacity of peers may be much more restrictive than the upload capacity of media servers located on the Internet; as thse peers are usually connected via bandwidth-constrained wireless links. On the other hand, if the load is evenly distributed among peers, serving the media content from lots of peers provides scalability and can

increase system throughput. Another issue is that the connection between an uploading peer and a downloading peer may not be stable during the duration of the streaming session, due to node mobility or peer disconnections [68]. Additionally the download path that is going over multiple peers may cause additional delay and increased jitter.

P2P streaming can utilize multiple peers as the sources of the same media file. As a result, if there are N such source peers, then each one will require R/N upload capacity where R is the streaming rate. Additionally, a peer that has downloaded the content may start serving the content to other peers, in this way increasing the number of serving peers.

The characteristics of wireless multi-hop networks require modifications of existing P2P media applications to run efficiently. For example, in [43], the authors propose a new set of criterias and methods to select super-peers in a P2P network providing IP telephony service. For ad-hoc networks, the selection criteria depend not only on the CPU, memory and storage capabilities of candidate super-peers, but also on the location of super-peers, their accessibility and their distance to other super-peers.

6 Summary

In this chapter, we have investigated the opportunities and challenges of the application of peer-to-peer concepts to mobile ad-hoc networks. An overview of P2P overlay networks shows that unstructured P2P systems do not impose a rigid relation between the overlay topology and resource locations, representing an easy implementation for dynamic environments such as MANETs. DHTs impose a structure on the overlay topology by satisfying certain criteria depending on the respective DHTs. An overview of mobile ad-hoc networks characteristics shows that mobile ad-hoc networks impose several problems in terms of wireless multi-hop characteristics leading to high and varying packet loss and delay, caused by collisions and interference among nodes. Future challenges such as disrupted communications, and intermittent connectivity in these scenarios are also envisioned. Most of the relevant schemes of MANET routing protocols are also briefly presented, giving focus on flat routing approaches such as topology-based, geographic-based, and probabilistic routing.

Although there is an inherent similarity, common peer-to-peer systems must be modified in many ways to enable their use in ad-hoc networks. Several approaches improve the performance of unstructured and structured P2P communication in wireless multi-hop networks. Meanwhile, different principles, such as layered, integrated and cross-layered design, guide to different integration and interaction possibilities between the peer-to-peer layer and the network layer. According to the simulation results, the deployment of a P2P protocol as is on top of ad-hoc routing layer cause significant message overhead and redundancy in communication. Thus, the integrated and cross-layered designs for unstructured P2P are shown to be intuitive and simple as modifications to the ad-hoc routing protocols are minimal. However, the network-wide broadcast of ad-hoc routing due to search requests (reactive) or topology change (proactive) does not scale to neither growing network sizes nor network mobility. In order to overcome that, some proposals push the DHT concept to the ad-hoc routing layer, enabling key-based routing for MANETs. Moreover, some of them explicitly considers physical locality in order to construct the overlay, while trying to keep minimum overhead.

As peer-to-peer applications gain greater importance in the infrastructure Internet, efficient porting of such applications to wireless scenarios is also discussed. Therefore, the solutions presented in Session V pave the way to the deployment of distributed applications such as decentralized name service, overlay-based multicast, multimedia service, and several other possibilities.

Acknowledgements This work was supported by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract n. 216715)

References

1. S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371, 2004.
2. I. Baumgart. P2pns: A secure distributed name service for p2psip. In *Proc. of Mobile P2P*, 2008.
3. B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, July 1970.
4. I. Bouazizi. Ara - the ant-colony based routing algorithm for manets. In *Proc. of ICPPW*, Washington, DC, USA, 2002. IEEE Computer Society.
5. J. Burgess, B. Gallagher, D. Jensen, and B. Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In *Proc. of INFOCOM*, Vancouver, Canada, August 2006.
6. S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, 2003.
7. S. Burresi, C. Canali, M. E. Renda, and P. Santi. Meshchord: A location-aware, cross-layer specialization of chord for wireless mesh networks. In *Proc. of PerCom*, Hong Kong, 2008.
8. M. Caesar, M. Castro, E. B. Nightingale, and G. OShea. Virtual ring routing: Network routing inspired by dhts. In *Proc. of ACM SIGCOMM*, Pisa, Italy, September 2006.
9. J. Cao, Y. Zhang, G. Cao, and L. Xie. Data consistency for cooperative caching in mobile environments. *IEEE Computer*, 37:60–66, april 2007.
10. Pietro Manzoni-Manuel P. Malumbres Carlos Miguel Tavares Calafate, Juan-Carlos Cano. A qos architecture for manets supporting real-time peer-to-peer multimedia applications. In *ISM*, 2005.
11. M. Castro, M. B. Jones, A-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman. An evaluation of scalable application level multicast built using peer-to-peer overlays. In *Proc. of INFOCOM*, 2003.
12. M. C. Castro, E. Villanueva, I. Ruiz, S. Sargento, and A. J. Kassler. Performance evaluation of structured p2p over wireless multi-hop networks. In *Proc. of MESH*, 2008.
13. V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-tolerant networking architecture. http://www.ietf.org/rfc/rfc4838.txt. IETF RFC 4838.
14. T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol. In *Proc. of the IEEE INMIC*, 2001.
15. M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross-layering in mobile ad hoc network design. *Computer*, 37(2):48–51, 2004.
16. M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross layering in mobile ad hoc network design. In *IEEE Computer*, 2004.
17. C. Cramer and T. Fuhrmann. Performance evaluation of chord in mobile ad hoc networks. In *Proc. of MobiShare*, 2006.
18. F. Delmastro. From pastry to crossroad: Cross-layer ring overlay for ad hoc networks. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 60–64, 2005.
19. F. Delmastro, A. Passarella, and M. Conti. P2p multicast for pervasive ad hoc networks. *Pervasive Mob. Comput.*, 4(1):62–91, 2007.
20. K. K. Dhara, V. Krishnaswamy, and S. Baset. Dynamic peer-to-peer overlays for voice systems. In *Proc. of IEEE International Conference on Pervasive Computing and Communications Workshops*, March 2006.
21. G. Ding and B. Bhargava. Peer-to-peer file-sharing over mobile ad hoc networks. In *Proc. of PERCOMW*, 2004.

22. D. E. Eastlake and P. E. Jones. Us secure hash algorithm 1 (sha1). http://www.ietf.org/rfc/rfc3174.txt. IETF RFC 3174.
23. M. Fiore, C. Casetti, and C.-F. Chiasserini. Efficient retrieval of user contents in manets. In *Proc. of INFOCOM*, Anchorage, AK, USA, May 2007.
24. T. Fuhrmann, P. Di, K. Kutzner, and C. Cramer. Pushing chord into the underlay: Scalable routing for hybrid manets. Technical report, Universitt Karlsruhe (TH), June 2006.
25. M. Gerla, C. Lindemann, and A. Rowstron. P2p manet's - new research issues. 2005.
26. I. Gruber, R. Schollmeier, and W. Kellerer. Performance evaluation of the mobile peer-to-peer protocol. In *Proc. of GP2PC*, 2004.
27. T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *Proc. of INFOCOM*, pages 1568–1576, 2001.
28. T. Hara. Replica allocation methods in ad hoc networks with data update. *Mobile Networks and Applications*, 8(4), 2003.
29. O. Heckmann and A. Bock. The edonkey2000 protocol. Technical report, Darmstadt University of Technology, December 2002.
30. C. Hoh and R. Hwang. P2p file sharing system over manet based on swarm intelligence: A cross-layer design. In *Proc. of WCNC*, March 2007.
31. G. Holland and N. H. Vaidya. Analysis of tcp performance over mobile ad hoc networks. In *Proc. of MobiCom*, pages 219–230, August 1999.
32. Y. Charlie Hu, Saumitra M. Das, and Himabindu Pucha. *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, chapter Peer-to-Peer Overlay Abstractions in MANETs, pages 858–871. CRC Press, 2005.
33. D. B. Johnson. Routing in ad hoc networks of mobile hosts. In *Proc. of the IEEE Workshop on Mobile Computing Systems and Applications*, pages 158–163, Santa Cruz, USA, December 1994.
34. D. B Johnson and D. A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.
35. Murat Karakaya, Ibrahim Korpeoglu, and Ozgur Ulusoy. Free riding in peer-to-peer networks. In *IEEE Internet Computing, to appear, 2009*.
36. B. N. Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proc. of MobiCom*, pages 243–254, August 2000.
37. W. Kellerer and R. Schollmeier. Proactive search routing for mobile peer-to-peer networks: Zone-based p2p. In *Proc. of ASWN*, 2005.
38. A. Klemm, C. Lindemann, and O. P. Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *Proc. of IEEE VTC*, October 2003.
39. E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. of CCCG*, pages 51–54, Vancouver, Canada, August 1999.
40. J. Liang, R. Kumar, and K. Ross. Understanding kazaa. http://citeseer.ist.psu.edu/liang04understanding.html, 2004.
41. A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *ACM Mobile Computing and Communications Review*, 7(3):19–20, 2003.
42. E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay networks schemes. *IEEE Communications Surveys and Tutorials*, 7(2):72–93, 2004.
43. N. Crespi M. Mani, W. Seah. Super nodes positioning for p2p ip telephony over wireless ad hoc networks. In *MUM*, 2007.
44. L. F. Cranor M. Waldman, A. D. Rubin. Publius: a robust, tamper-evident, censorship-resistant web publishing system. In *Proc. of USENIX*, Denver, Colorado, USA, 2000. ACM Press.
45. V. Naumov, R. Baumann, and T. Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proc. of ACM MobiHoc*, pages 1568–1576, Florence, Italy, May 2006.
46. A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich. Better approach to mobile ad-hoc networking (b.a.t.m.a.n.). http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00, April 2008. work in progress.

47. M. Papadopouli and H. Schulzrinne. A performance analysis of 7ds: a peer-to-peer data dissemination and prefetching tool for mobile users. In *Proc. of IEEE Advances in Wired and Wireless Communications*, March 2001.
48. C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proc. of IEEE WMCSA*, February 1999.
49. C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Proc. of the IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
50. C. Plaxton, R. Rajaraman, and A. Richa. Accessing nearby copies of replicated objects in a distributed environment. In *Proc. of ACM SPAA*, 1997.
51. J. Pouwelse, P. Garbacki, D. Epema, and H. Sips. The bittorrent p2p file-sharing system: Measurements and analysis. *Peer-to-Peer Systems IV*, pages 205–216, 2005.
52. H. Pucha, S. M. Das, and Y. Charlie Hu. Ekta: An efficient dht substrate for distributed applications in mobile ad hoc networks. In *Proc. of the Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2004)*, English Lake District, UK, December 2004.
53. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. of ACM SIGCOMM*, San Diego, USA, August 2001.
54. S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling churn in a DHT. In *Proc. of USENIX*, June 2004.
55. M. Ripeanu. Peer-to-peer architecture case study: Gnutella network. In *Proc. of Peer-to-Peer Computing*, pages 99–100, 2001.
56. David Molnar Roger Dingledine, Michael J. Freedman. The free haven project: Distributed anonymous storage service. In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability, LNCS*, 2000.
57. A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of ACM/IFIP Middleware*, Heidelberg, Germany, November 2001.
58. R. Schollmeier, I. Gruber, and M. Finkenzeller. Routing in mobile ad hoc and peer-to-peer networks: a comparison. In *Proc. of Workshop on Peer-to-Peer Computing. In Networking*, 2002.
59. R. Schollmeier, I. Gruber, and F. Niethammer. Protocol for peer-to-peer networking in mobile environments. In *Proc. of ICCCN*, 2003.
60. H. Sozer, M. Tekkalmaz, and I. Korpeoglu. A peer-to-peer file search and download protocol for wireless ad-hoc networks. *to appear in Computer Communications*, To Appear.
61. I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of ACM SIGCOMM*, San Diego, USA, August 2001.
62. I. Stojmenovic. Position based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, 2002.
63. B. Tang, H. Gupta, and S. Das. Benefit-based data caching in ad hoc networks. *IEEE Trans. on Mobile Computing*, 7(3):62–91, 2008.
64. A. Varshavsky, B. Reid, and E. de Lara. A cross-layer approach to service discovery and selection in manets. 2005.
65. B. Wiberg. *Porting aodv-uu implementation to ns2 and enabling tracebased simulation*. Master's thesis, Uppsala University, 2002.
66. R. Winter, T. Zahn, and J. Schiller. Random landmarking in mobile, topology-aware peer-to-peer networks. In *Proc. of FTDCS*, 2004.
67. O. Wolfson, B. Xu, H. Yin, and H. Cao. Search-and-discover in mobile p2p network databases. In *Proc. of Int. Conf. on Distributed Computing Systems*, 2006.
68. Qian-Ni Deng Jin-Yuan You Xue Guangtao, Ming-Lu Li. Stable group model in mobile peer-to-peer media streaming system. In *First IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2004.
69. L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. *IEEE Trans. on Mobile Computing*, 5(1), 2006.

70. T. Zahn. *Structured Peer-to-Peer Services for Mobile Ad Hoc Networks*. Phd thesis, Freien University Berlin, 2006.
71. T. Zahn and J. Schiller. Mapnas: A lightweight, locality-aware peer-to-peer based name service for manets. In *Proc. of LCN*, pages 499–500, Washington, DC, USA, 2005. IEEE Computer Society.
72. T. Zahn and J. Schiller. Dht-based unicast for mobile ad hoc networks. 2006.
73. Y. Zhu, J. Rexford, A. Bavier, and N. Feamster. Ufo: A resilient layered routing architecture. In *Technical Report TR-780-07, Princeton University*, 2007.