47

Peer-to-Peer Overlay Abstractions in MANETs

Y. Charlie Hu, Saumitra M. Das,
and Himabindu Pucha

47.1 Introduction

A mobile ad hoc network (MANET) consists of a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration. In such a network, nodes operate as both end hosts and routers, forwarding packets for 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. The applications enabled by such collaborations often require services for locating resources, sharing information and data, multicasting, etc. In addition, due to the infrastructure-less environment, these applications should be designed to operate in a decentralized manner.

Recently, peer-to-peer (P2P) systems consisting of a dynamically changing set of nodes connected via the Internet have gained tremendous popularity. While initially conceived and popularized for the purpose of file sharing (for example, Gnutella and Freenet,[1,15]). P2P has emerged as a general paradigm for the construction of resilient, large-scale, distributed services and applications in the Internet. We broadly define P2P systems as self-organizing, decentralized distributed systems that consist of potentially untrusted, unreliable nodes with symmetric roles. The self-organization, decentralization, diversity, and redundancy inherent in the approach are relevant to a large class of applications beyond file sharing, anonymity, and anti-censorship.

The P2P paradigm has largely adopted a layered approach. A P2P overlay network built on top of the Internet provides a general-purpose substrate that provides many common properties desired by distributed applications, such as self-organization, decentralization, diversity, and redundancy. Such an overlay shields distributed application designers from the complexities of organizing and maintaining a secure overlay, tolerating node failures, balancing load, and locating application objects.

While largely developed independently of each other, P2P overlay networks in the Internet and mobile wireless ad hoc networks share many key characteristics such as self-organization and decentralization due to the common nature of their distributed components: a P2P overlay network consists of a dynamically changing set of nodes connected via the Internet, and a mobile ad hoc network consists of mobile nodes communicating with each other using multi-hop wireless links. These common characteristics lead to further similarities between the two types of networks: (1) both have a flat and frequently changing topology, caused by node join and leave in P2P overlays and MANETs and additionally terminal mobility of the nodes in MANETs; and (2) both use hop-by-hop connection establishment. Per-hop connections in P2P are typically via TCP links with physically unlimited range, whereas per-hop connections in MANETs are via wireless links, limited by the radio transmission range.

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, 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. The common characteristics and design goals between P2P overlays and mobile ad hoc networks point to a new research direction in networking, that is, to exploit the synergy between P2P overlays and mobile ad hoc networks to design better routing protocols and applications.

47.1.1 Scope of Chapter

We use Figure 47.1 to define the scope of this chapter. Figure 47.1 depicts the network protocol design space for the Internet and MANETs as well as example protocols for each design subspace.

The protocols developed for the Internet can be broadly classified into two categories:

1. Protocols that have been developed using a router-assisted approach. Examples of such protocols include Internet routing protocols such as RIP,[25] OSPF,[45] and BGP[54] and IP multicast protocols such as MOSPF[44] and DVMRP.[21]

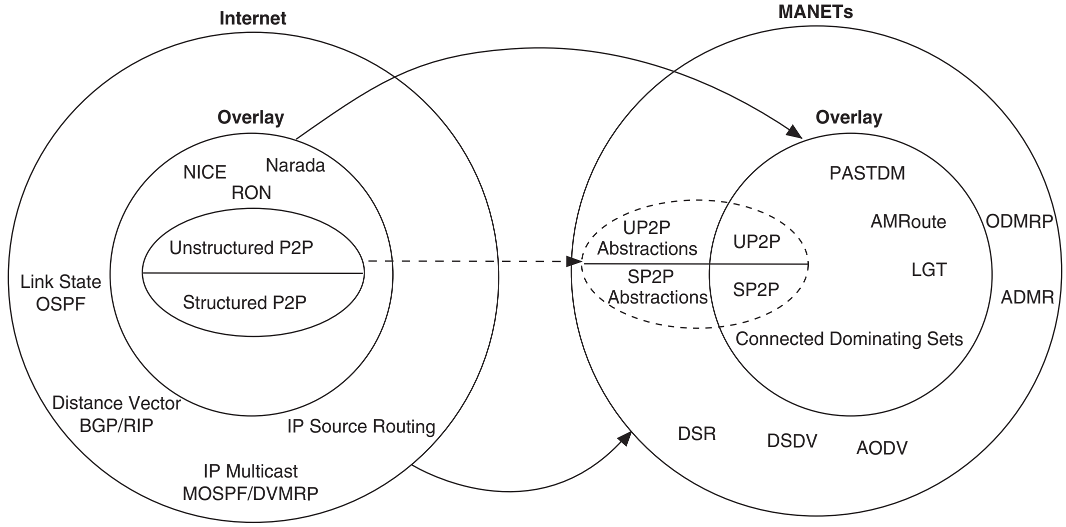

FIGURE 47.1 Scope of this chapter: how to map unstructured and structured P2P overlays in the Internet to MANETs. S-P2P refers to structured P2P and U-P2P refers to unstructured P2P.

2. Protocols that have been developed for an overlay of nodes (depicted by the inner circle in Figure 47.1), which typically refer to end hosts on the edge of the network.

P2P research in the Internet has concentrated on overlay networks of end hosts. The set of overlay-based protocols includes unstructured P2P protocols (e.g., Gnutella,[1] KaZaA[33]) and structured P2P protocols (e.g., Chord,[61] Tapestry,[71] Pastry,[56] CAN[51]). Examples of other overlay-based protocols include RON,[4] Narada,[26] and NICE.[5]

Similarly, MANETs have protocols parallel to those in the Internet domain. In MANETs, some approaches involve all the nodes in the network. These include routing protocols such as DSR[30] and AODV[50] as well as multicast protocols such as ADMR[28] and ODMRP.[37] On the other hand, some protocols (e.g., overlay-based multicast such as AMRoute,[69] virtual infrastructures[17,40]) have been developed to operate on an overlay (subset) of nodes in MANETs.

The scope of this chapter is on mapping the unstructured and structured P2P protocols developed for the Internet to the MANET protocol design space. Such a mapping raises several key questions:

- Are there any benefits of this mapping for designing new protocols and applications in MANETs?
- Should these P2P overlay protocols be mapped to an overlay or a non-overlay (the whole network) in MANETs?
- How should these P2P overlay protocols be adapted for MANETs?

In the remainder of the chapter, we survey recent studies on mapping unstructured and structured P2P overlay protocols in the Internet to MANETs. We first survey studies investigating how to efficiently support P2P overlay abstractions in MANETs. We then discuss how and when applications in MANETs can benefit from the use of P2P abstractions and examine the trade-offs involved in their operations.

We emphasize that due to the peer-to-peer nature of MANETs, all protocols designed for MANETs are inherently peer-to-peer. Examples include multi-hop routing protocols (e.g., DSR and AODV), multicast routing protocols (e.g., ADMR and ODMRP), overlay based multicast routing protocols (e.g., AMRoute and PAST-DM,[23] as well as numerous gossip-based data dissemination protocols (e.g., 7DS[48] and iFlow[39]). The scope of this chapter is focused on the P2P overlay protocols and applications that have been recently developed for the Internet and evaluates their usability and applicability for MANETs, as shown by the dashed arrow and dashed oval in Figure 47.1.

The rest of the chapter is organized as follows. Section 47.2 gives a brief overview of peer-to-peer overlay networks developed for the Internet. Section 47.3 discusses the motivation and challenges in supporting P2P overlay abstractions in MANETs. Section 47.4 and Section 47.5 survey recent work on supporting unstructured and structured P2P overlay abstractions in MANETs, respectively. Section 47.6 discusses potential applications of P2P overlay abstractions in the MANETs. Section 47.7 briefly summarizes other overlay-based techniques in MANETs. Section 47.8 concludes with a set of challenges and problems requiring further research.

47.2 Background on P2P Overlay Networks

The numerous P2P overlay networks for the Internet that have been proposed in the past few years can be broadly classified into two categories:

1. *Unstructured.* Unstructured P2P overlay networks as exemplified by Gnutella[1] do not have precise control over the overlay topology. The network is typically formed by nodes joining the network following some loose rules, for example, a node joining a Gnutella network starts by connecting to nodes in a *host cache* file which stores Gnutella nodes learned from the last time the node was part of a Gnutella network,[2] and a Gnutella node typically specifies a default maximal number of neighbors in the Gnutella overlay. The resulting network topology follows certain patterns, for example power-law-like,[42] but the placement of an object or a file is not based on any knowledge about the topology. Furthermore, the overlay is often not network proximity aware; that is, neighboring nodes in the overlay may be far away from each other in the underlying Internet topology. The typical way of locating an object in an unstructured overlay is to flood the network in which a query is propagated to overlay neighbors within a controlled radius. We note more efficient ways of locating objects in unstructured overlays exist (for example, Ref. 12) and a detailed discussion is beyond the scope of this survey. While the lack of proximity-awareness and flooding-based object location are inefficient, the consequent advantage is that unstructured overlay networks and the companion object location mechanisms that do not rely on any precise structure of the topology are highly resilient to frequent node join and departure.
2. *Structured.* To overcome the inefficiency with object location in unstructured networks, structured overlay networks have been proposed to combine the inherent self-organization, decentralization, and diversity of unstructured P2P overlays with a scalable and efficient routing algorithm that can reliably locate objects in a bounded number of routing hops, typically logarithmic in the network size, while exploiting proximity in the underlying Internet topology. Numerous structured P2P overlays have been proposed, such as CAN,[51] Chord,[61] Pastry,[56] and Tapestry.[71] The routing of such structured P2P overlays effectively implements scalable and fault-tolerant *distributed hash tables* (DHTs): each node in the network has a unique node identifier (nodeID) and each data item stored in the network has a unique key, nodeIDs and keys live in the same namespace, and a message with a key is routed (mapped) to a unique node in the overlay.* Thus, DHTs allow data to be inserted without knowing where it will be stored and requests for data to be routed without requiring any knowledge of where the corresponding data items are stored. To maintain efficient routing, nodes in a structured overlay must maintain neighboring nodes that satisfy certain criteria in the namespace. As a result, structured overlays are conceptually less resilient to frequent node join and departure.

*In the rest of the chapter, we use DHTs and structured P2P overlays interchangeably whenever appropriate for this reason.

47.3 Supporting P2P Overlay Abstractions in MANETs

In this section we first elaborate the difference between supporting P2P overlay abstractions and P2P overlays in MANETs. We then discuss the motivations for and challenges in supporting P2P overlay abstractions in MANETs. Finally, we give a taxonomy of different design approaches to supporting P2P overlay abstractions in MANETs.

47.3.1 Why P2P Overlay Abstractions and not P2P Overlays?

Although an overlay consisting of a subset of the nodes could be constructed in an ad hoc network, similarly to in the Internet, all studies of P2P overlays in MANETs have assumed the involvement of all nodes. This is because an ad hoc network is typically formed of nodes that collaborate with each other to enable communication among all the nodes. Because all nodes are involved, the notion of overlay in the Internet (i.e., consisting of a subset of end hosts) is no longer precise. What the existing studies really support is effectively a *P2P overlay abstraction*, that is, borrowing the topologies and objection location techniques (for example, DHTs) of P2P overlays developed in the Internet and supporting them in MANETs. In the rest of the chapter, we focus on P2P overlay abstractions in MANETs, shown as “U-P2P abstractions” and “S-P2P abstractions” in Figure 47.1.

Because overlays in the Internet are built to circumvent the fact that router-assisted approaches are not feasible, message routing is done in the application layer overlay. In contrast, because nodes in MANETs are end hosts as well as routers, all nodes in MANETs are effectively involved in supporting P2P overlay abstractions, and thus P2P overlay abstractions in MANETs have the option of being implemented either at the network layer or above, that is, at the application layer. These design options are discussed further in Section 47.3.4.

47.3.2 Why Support P2P Overlay Abstractions in MANETs?

The motivation for supporting P2P overlay abstractions in MANETs is that MANETs and P2P overlays share many essential characteristics such as decentralization that make P2P applications developed in the Internet potential candidates for deployment in MANETs. For example, P2P file-sharing applications such as Gnutella are designed with a serverless architecture, which makes them potentially well suited to the infrastructure-less MANET environment.

Structured P2P overlays developed for the Internet have been shown to provide a general substrate for building a variety of scalable and robust distributed applications for the Internet, such as distributed storage systems,[16,55] application-level multicast,[10,11,52,70,72] and content-based full-text search[64]. A DHT abstraction implemented by these structured P2P overlays shields many difficult issues, including fault tolerance, locating objects, scalability, availability, load balancing, and incremental deployment from the distributed application designers. The motivation for supporting the DHT abstraction in MANETs is similar to its counterpart in the Internet. Due to its support for many properties common to distributed applications, a DHT abstraction, if deployed in MANETs, could similarly shield many complexities in constructing distributed applications from the application designers. For example, applications such as file sharing and resource discovery could benefit from the distributed insert/lookup convergence provided by DHTs.

47.3.3 Challenges in Supporting P2P Overlay Abstractions in MANETs

Many fundamental differences between the Internet and a mobile ad hoc network pose challenges to implementing P2P overlay abstractions in MANETs, including:

- *Bandwidth limitation.* Unlike the wired Internet, MANETs have lower network capacity due to the use of wireless channels. This limits the usability of P2P protocols that have high message overhead.

- *Multi-access interference.* Multiple access techniques such as CSMA/CA are required for nodes in a MANET to acquire the wireless channel and transmit data. Because no central coordination point exists in a MANET, collisions and delays occur in acquiring the wireless channel. These problems can be aggravated further by P2P applications that rely on high message overhead mechanisms such as frequent pings for delay estimation.
- *Node mobility.* In the Internet, the topology of a P2P overlay changes at a large time scale. On the other hand, in a MANET, limited transmission range and node mobility results in frequent topology changes. This places pressure on P2P applications constructed in MANETs to update the overlay topology much more frequently to maintain the matching between the overlay topology and the underlying network topology.

  Topology maintenance in P2P overlays is achieved by periodically probing current and candidate neighbors and selecting closer nodes as new neighbors. In the Internet, such an approach is feasible because routes between overlay neighbors change rarely; the maintenance is more for the purpose of checking whether a neighbor is alive. Even if the routes do change, there is little overhead for discovering a new route to an overlay neighbor in the Internet. In MANETs, due to limited link capacity and multi-access interference, probing an overlay neighbor can be much more costly.
- *Churn.* In the Internet, structured P2P protocols are particularly affected by "churn" (frequent node joining and leaving the network). This occurs in the Internet primarily because most nodes in the systems are end-user desktops rather than "always-on" servers. If structured P2P protocols are used in MANETs, they could potentially suffer poor performance due to high churn caused by not only the transience of node in terms of being "on" and "off," but also network partition (un-reachability) caused by node mobility.
- *Lack of infrastructure.* Certain P2P protocols make use of some infrastructure components in their designs. For example, a P2P routing protocol may assign node identifiers based on locations determined from static landmarks to improve routing performance.[51] These techniques may not be usable in MANETs due to the lack of any static infrastructure.
- *Limited energy.* Most P2P applications in the Internet are not designed to operate with minimum message transmissions. In an energy-limited environment such as a MANET, it may be very important for nodes to reduce the number of message transmissions while keeping the performance acceptable. For example, most P2P protocols use proactive maintenance of state information (e.g., periodically exchanging and probing routing table entries to maintain the proximity of routing tables in Pastry[9]). In a MANET, reactive approaches exemplified by protocols like AODV[50] and DSR.[30] in which the protocol state is refreshed only when required, may be more effective.
- *State-efficiency trade-off.* Structured P2P routing protocols trade increased number of routing hops (in the overlay) for dramatically reduced state in order to scale to millions of nodes. In a MANET, it is an open question whether such a large number of nodes can ever effectively be organized into one infrastructure-less network. Because MANETs are likely to be much smaller than overlays in the Internet and have much smaller capacity compared to the Internet, it may be more effective to keep more state at each node if it can reduce the number of hops traveled by messages.
- *Addressing.* Nodes in a MANET are likely to disconnect and reconnect to the network many times. Although no specific addressing architecture has been standardized for MANETs, it is plausible to assume that nodes will have changing IP addresses over time. This could challenge structured P2P protocols that store logical to physical address (nodeID-to-IP) mappings in their routing tables. In some cases, nodeIDs are assigned as hashed IP addresses, and thus the nodeIDs of MANET nodes will continuously change. This can result in consistency issues for structured P2P protocols. If nodeIDs are assigned by hashing static MAC addresses, then the nodeID-to-IP mappings of nodes could change over time.
- *Namespace versus physical space routing.* Structured P2P protocols route packets in a logical namespace, that is, based on nodeIDs. The routing in the logical namespace could be tree-like (prefix based)[56,71] or skiplist-like.[61] Although prefix-based routing allows for natural inclusion of topology

awareness, that is, selecting physically nearby logical nodes with matching prefix nodeIds in the namespace at each routing step, the routes going through the overlay are unavoidably longer than the direct routes in the underlying network. Thus, this "route stretch," along with the inflexibility in the selection of intermediate overlay nodes, could cause inefficiency for certain applications in MANETs, such as multicast.

47.3.4 A Taxonomy of Design Approaches

The various design approaches for applying peer-to-peer to MANETs can be classified based on the nature of the P2P overlays (structured versus unstructured). In addition, as discussed in Section 47.3.1, because nodes in MANETs are end hosts as well as routers and all nodes in MANETs are effectively involved in supporting P2P overlay abstractions, the P2P overlay abstraction in MANETs has the option of being implemented either at the network layer or above, that is, at the application layer. If a protocol is layered over an existing routing protocol for MANETs (DSR,[30] AODV,[50] a etc.), we classify it as the *layered design*. If a protocol is integrated with a MANET routing protocol at the network layer, we classify it as the *integrated design*.

The layered design allows P2P applications developed in the Internet to be easily ported to MANETs. Furthermore, it decouples functionalities of the application layer (P2P application) and the network layer (routing), which enables independent development of protocols at the two layers. However, MANETs are a limited resource environment where the performance can be more important than portability and separation of functionalities. In fact, a multitude of cross-layer techniques have been proposed for MANETs for this reason. Thus, for supporting P2P overlay abstractions in MANETs, it is important to study the potential benefits of cross-layering, for example, in an integrated design.

The four design approaches to supporting P2P overlay abstractions in MANETs are:

1. *Layered and unstructured.* In this design, off-the-shelf unstructured P2P protocols (for example, Gnutella) are operated on top of an existing MANET routing protocol. This design is similar to the approach in the Internet, which layers a P2P protocol on top of the existing IP infrastructure.
2. *Integrated and unstructured.* In this design, the operation of an unstructured P2P protocol is closely integrated with the operation of a MANET routing protocol to support unstructured P2P APIs.
3. *Layered and structured.* In this design, a structured P2P protocol such as Pastry, CAN, Chord, or Tapestry runs as an application over a MANET routing protocol.
4. *Integrated and structured.* In this design, a structured P2P protocol is integrated with the operations of a MANET routing protocol to provide a distributed hash table abstraction.

47.4 Unstructured P2P Overlay Abstractions in MANETs

47.4.1 Layered Design

The work Oliveira et al.[47] studies the performance of an unstructured P2P application in a MANET running over three existing MANET routing protocols. Specifically, the authors studied the relative performance of DSR, AODV, and DSDV when supporting an unstructured P2P application based on Gnutella. The results observed show that the performances of the protocols are different from those observed for typical unicast applications assumed in previous studies.[8,20] In addition, the packet delivery ratios observed are lower than those observed for typical unicast applications.

The disadvantage of the layered unstructured design is explained by considering a query search for a data item, (e.g., as in Gnutella). Similar to in the Internet, in this design, Gnutella operating at the application layer will contact the overlay neighbors to resolve the query. These overlay neighbors will further forward the query to their neighbors if the decremented TTL of the query is greater than zero. However, due to node mobility, these overlay neighbors may not reflect the current physical topology of the ad hoc network, and thus may need a multi-hop route to be reached. As a result, each such overlay hop required by Gnutella at the application layer could result in a costly flooding-based route discovery by the multi-hop routing protocol. This suggests strong motivation for continuous update of the list of overlay neighbors to reflect the current physical neighbors so that neighbors can be contacted via one-hop physical broadcast rather than multi-hop routing.

47.4.2 Integrated Design

The work Klemm et al.[34] proposes integrating an unstructured Gnutella-like P2P application into the network layer and compares it to a layered design similar to that of Oliveira et al.[47] The proposed file-sharing application, ORION, allows for the setup of on-demand overlay connections that closely match the physical topology of the underlying MANET. ORION integrates the query process required for the P2P operation with the routing process done by the routing protocols in MANETs. Specifically, ORION combines the P2P operation with routing techniques from AODV[50] and the Simple Multicast and Broadcast Protocol.[27] When a query for a data item arrives, ORION employs one-hop broadcast to contact all its physical neighbors in one transmission. These nodes, in turn, contact their physical neighbors if they are unable to resolve the query. Nodes that can resolve the query stop further propagation of the query (i.e., one-hop broadcast) and reply to the initiator of the query. Note that this may require multi-hop routing. However, such a route has already been set up as the query is propagated in the network, similar to that in AODV.

47.4.3 Comparison

The advantage of the integrated design of ORION for a file-sharing application was experimentally observed by Klemm et al.[34] ORION is implemented in the ns-2 simulator[7] and compared to an off-the-shelf design that layers Gnutella over DSR. The simulations are carried out for a range of network sizes up to 60 nodes using a random waypoint mobility model in which nodes move at a maximum speed of 2 m/s with a pause time of 50 seconds. The search accuracy (i.e., the fraction of received unique files out of all the files actually in the network that match the search query) is measured for both designs. The results show that as the network size is increased from 0 to 60 nodes, the search accuracy of the integrated design increases and is always higher than that of the layered design. In fact, the accuracy for the layered design decreases at large network sizes due to increased overhead. The layered design experiences a large fraction of total packet transmissions related to control traffic.

In summary, the primary factor for the worst performance of the layered design is not the inefficiency of the search mechanism, but rather the overhead of maintaining static overlay neighbor connections that are not adapted to the dynamic physical topology. The results of the study indicate that the integrated unstructured design has significantly lower overhead compared to the layered design while achieving better performance according to application-specific metrics.

47.5 Structured P2P Overlay Abstractions in MANETs

We separate the discussion of supporting DHTs in wireless ad hoc networks with and without GPS support. With GPS support (or some other type of position services), each node in the network can find out its geographic location. Previously, this location information was exploited to improve the efficiency and scalability of multi-hop routing protocols[43] or support other types of services such as geocast.[46] Such location information can also be potentially used to improve the efficiency of DHTs.

47.5.1 DHTs in MANETs without GPS Support

Both the layered design and the integrated design in supporting a DHT abstraction in highly dynamic mobile ad hoc networks are explored in Ref. 19. In particular, the simple design of directly overlaying a DHT on top of an existing multi-hop routing protocol for MANETs is compared with Ekta, which integrates a DHT with a multi-hop routing protocol at the network layer.

47.5.1.1 Layered Design

In the layered design, a proximity-aware DHT, Pastry,[9,56] is directly layered on top of a multi-hop routing protocol, DSR,[30] with minimum modifications to the routing protocol. Pastry maintains its leaf set and routing table entries without source routes and DSR maintains source routes passively as per the demand of the Pastry routing state.

However, a straightforward layering is not pragmatic, and three modifications are made to accommodate the shared medium access nature of MANETs: (1) Pastry's node joining process is modified to use expanding ring search in locating a bootstrap node to join the network; (2) the original Pastry uses an expensive "ping" mechanism with a delay metric to measure and maintain the proximity of nodes in its routing tables; pastry is modified to use a hop count metric for proximity because in MANETs, delay is affected by many factors and has a high variability; (3) to reduce the cost of this proximity probing, DSR is modified to export an API that allows Pastry to inquire about the proximity values for nodes in which it is interested. DSR can then use its cache to reply to "pings" from Pastry if there is a cached path to the node being pinged.

In summary, the layered design is similar to implementing a DHT in the Internet; it leverages the existing routing infrastructure for MANETs to the fullest extent. This design, while consistent with the layered principle of the ISO model of networking, makes it difficult to exploit many optimization opportunities from the interactions between the DHT protocol and the underlying multi-hop routing protocol. For example, when the routing protocol is Dynamic Source Routing (DSR)[30] which uses caching to reduce the routing overhead, it is difficult for the routing structures of the DHT and the route cache of DSR to coordinate with each other to optimally discover and maintain source routes.

47.5.1.2 Integrated Design

In the integrated design, called Ekta by Das et al.,[19] the functions performed by the Pastry DHT protocol operating in a logical namespace are fully integrated with the MANET multi-hop routing protocol DSR operating in a physical namespace. The key idea of the integration is to bring the DHT routing protocol of Pastry to the network layer of MANETs via a one-to-one mapping between the nodeIDs of the mobile nodes in the namespace and their IP addresses. With this integration, the routing structures of a DHT and of a multi-hop routing protocol (e.g., the route cache of DSR) are integrated into a single structure that can maximally exploit the interactions between the two protocols to optimize the routing performance.

- *Node addressing.* Ekta assigns unique nodeIDs to nodes in a MANET by hashing the IP addresses of the hosts using collision-resistant hashing functions such as SHA-1.[22]
- *Node state.* The structures of the routing table and the leaf set stored in each Ekta node are similar to those in Pastry. The difference lies in the content of each leaf set and routing table entry. Because there is no underlying routing infrastructure in MANETs, each entry in the Ekta leaf set and routing table stores a nodeID and a source route to reach the designated node. As in Pastry, any routing table entry is chosen such that it is physically closer than the other choices for that routing table entry. This is achieved by making use of the vast amount of indirectly received routes (from overhearing or forwarded messages) as described below in optimizations.

  For efficiency, each routing table entry stores a vector of source routes to one or more nodes that match the prefix of that entry. Similarly, each leaf set entry stores multiple routes to the designated node. The replacement algorithm used in each leaf set or routing table entry is Least Recently Discovered (LRD), disregarding whether a route is discovered directly or indirectly. When looking up a route from a leaf set or routing table entry, the freshest among the shortest routes in that vector entry is returned.
- *Routing.* In Ekta, a message with a key is routed using Pastry's prefix-based routing procedure and delivered to the destination node whose nodeID is numerically closest to the message key. When a route lookup for the next logical hop returns a next-hop node from the leaf set for which a source route does not exist, Ekta initiates route discovery to discover a new source route. On the other hand, if the node selected as the next hop is from the routing table, a modified prefix-based route discovery is performed to discover routes to any node whose nodeID matches the prefix for that

routing table entry. Note that each hop in the Ekta network is a multi-hop source route, whereas each hop in a corresponding Pastry network is a multi-hop Internet route.

- *Optimizations.* Ekta inherits all of the optimizations on route discovery and route maintenance used by the DSR protocol. In addition, Ekta updates its routing table and leafset using routes snooped while forwarding and overhearing packets, thus constantly discovering fresh and low-proximity routes for the leaf set and the routing table entries. In addition to the "prefix-based view" of the routing table and the "neighbor-node view" of the leaf set, the Ekta routing structures can be viewed as two caches of source routes. These can be used to support unicast routing by Ekta whenever required by the application. For example, an *Insert* operation in a DHT-based application can travel over multiple hops in the nodeID space while an acknowledgment to the *Insert* could be efficiently unicast back to the originator.

47.5.1.3 Comparison

A detailed simulation study[19] was also performed to compare the layered design with the integrated design. The results show that the integrated design of supporting the DHT abstraction in MANETs used by Ekta is superior to the layered design in terms of the number of data packets successfully delivered and the average delay in delivering the packets while incurring comparable routing overhead. These results suggest that integrating the functionalities of the DHT into the routing layer is much more efficient than having two independent layers with minimal interactions in supporting a DHT abstraction in MANETs.

47.5.2 DHTs in MANETs with GPS Support

The geographic location system (GLS) in GRID[38] is a scalable location service that performs the mapping of a node identifier to its location. The implementation of GLS effectively provides a DHT abstraction: it routes a message with a message key Y to a node whose nodeID is closest to Y. However, the implementation of GLS itself requires both GPS support as well as building a distributed location database.

Geographic Hash Table (GHT)[53] is inspired by DHTs in the Internet but proposed to support the data-centric storage model in sensor networks. GHT hashes keys into geographic coordinates, and stores a key-value pair at the sensor node geographically nearest the hash of its key. That is, different from the objection location and routing in the namespace abstraction of DHT, GHT provides the distributed hashing abstraction directly in the geographic location space. Such an abstraction suits the data-centric storage model, in which a data object is stored on the unique node determined based on the object's name.

To implement GHT routing, GHT requires GPS support. In fact, GHT is implemented by extending the GPSR[32] geographic forwarding protocol with the notions of home node (the node nearest a hashed location) and home perimeter, which contains nodes enclosing a hashed location. To ensure persistence and consistency of stored key-value pairs in the presence of node failure and mobility, the stored key-value pairs are replicated among nodes on the home perimeter and are periodically refreshed.

47.6 Application of P2P Overlay Abstractions in MANETs

In this section we revisit distributed applications that have been built on top of P2P overlays in the Internet, and discuss whether it is suitable to build them on top of the P2P overlay abstractions supported in MANETs.

The design trade-offs involved when porting applications for unstructured P2P overlays depend on how the overlay topology maintained by the applications maps to the physical one. The most important modification that will be required of any unstructured P2P application deployed in MANETs is to reconcile the overlay neighbors with the physical neighbors continuously. Thus, the unstructured P2P applications would need to incorporate mechanisms for choosing physically nearby nodes as overlay neighbors, as well as techniques to gradually change the set of overlay neighbors as a node or its neighbors move.

There are subtle trade-offs involved when using a structured P2P (DHT) substrate for applications in MANETs. An efficient DHT substrate in MANETs such as Ekta can greatly ease the construction of distributed applications and services in MANETs by shielding many common and difficult issues such as fault tolerance, object location, load balancing, and incremental deployment from the developer. These issues are especially challenging in a wireless, mobile environment. Thus, providing a DHT substrate in MANETs is significant because it removes the need for developers to optimize each individual application in order to perform efficiently in MANETs. However, due to node mobility in MANETs, any DHT substrate based on on-demand routing can trigger repeated flooding-based route discoveries to discover and maintain routes. That is, an application built on top of DHT in MANETs may also experience many floodings of messages. Thus, the real question is whether applications built on top of the DHT substrate can be as efficient as or more efficient than those individually optimized to operate in MANETs.

47.6.1 File Sharing

File sharing is the most popular P2P application currently in use in the Internet. Due to the collaborative nature envisioned for MANETs, file sharing could potentially be one of the important applications for ad hoc networks. However, as discussed in Section 47.4, although Gnutella is widely used in the Internet, directly running applications like Gnutella over a routing protocol for MANETs could result in poor performance due to the mismatch between the overlay topology and the underlying network topology.

The work by Klemm[34] develops an integrated unstructured P2P Protocol (ORION) that works efficiently in MANETs. As discussed in Section 47.4, ORION uses one-hop broadcast flooding when searching for files in order to use the current physical neighbors instead of overlay neighbors. ORION also maintains a file-based routing table similar to AODV, with the only difference being that it stores next hops to reach files instead of other nodes in the network. Similar to a unicast MANET routing protocol, ORION caches information about the ownership of files learned from forwarded messages in its file routing table to improve routing performance. When a query for a file reaches a node, it can send a response if it stores that file or if it knows of a route to the file through its file routing table. Thus, responses to queries performed by a node result in widespread caching of file information throughout the network. This information can then be used to respond to future requests.

Additionally, the authors[34] use their own file transfer protocol different from TCP. The motivation for this approach is that the current sender of a file could change with network conditions, making an end-to-end approach like TCP inefficient. A file is split into equal-sized blocks prior to transfer. A file is fetched block by block by the querying node. This allows for parts of files to be fetched from different nodes based on the current network conditions. Because TCP is not used, ORION incorporates its own packet scheduling and loss-recovery mechanisms. File blocks can arrive out of order as long as one copy for each block is received.

The integrated design of a P2P file-sharing application (ORION) is experimentally compared to an off-the-shelf approach that layers Gnutella over DSR and AODV.[34] The simulations are carried out for a range of network sizes up to 65 nodes using a random waypoint mobility model in which nodes move at a maximum speed of 2 m/s with a pause time of 50 seconds. The number of successful file transfers is measured for 20,000 search queries, each for a file of size 3 Mb (the size of a typical MP3 file), for both designs. The results show that ORION is able to complete up to 40 percent more file transfers while using lower bandwidth compared to the layered design.

In principle, and similar to how the PAST[55] storage system is implemented on top of the Pastry DHT, a storage system in MANETs can be constructed on top of Ekta. However, the trade-offs involved and comparison with an unstructured integrated approach remain open issues.

47.6.2 Resource Discovery

Because wireless ad hoc networks are typically comprised of a wide variety of heterogeneous devices with varying energy resources, capabilities, and services to offer, such systems tend to rely on peer cooperation to efficiently use each other's resources. Examples of resource discovery include discovering nodes with GPS devices so other nodes can approximate their own locations, collecting sensed environmental information from mobile sensors, contacting location and directory servers, and locating people with specific capabilities in disaster relief and battlefield networks. For example, a resource lookup in a platoon of soldiers or in a team of coordinating disaster relief personnel can be of the form "*Find the closest medic.*" These examples show that efficient discovery of resources that meet certain requirements in an ad hoc network is of great importance in building a variety of distributed applications in ad hoc networks.

The work by Das et al.[19] first formally defines the resource discovery problem in mobile ad hoc networks and then presents two alternative designs for resource discovery: (1) an integrated structured solution built on top of Ekta (Ekta-RD), and (2) an integrated unstructured solution directly built on top of physical layer broadcast (DSR-RD). Both versions are modeled after the Service Location Protocol framework.[24]

In DSR-RD, the resource discovery application is integrated with the DSR routing protocol. It essentially uses physical layer broadcast augmented with source routing to perform resource discovery as follows. Each node transmits RESOURCE REQUEST packets (similar to ROUTE REQUEST packets of DSR), and each node that does not own the resource requested rebroadcasts the RESOURCE REQUEST packet after encoding its IP address into the source route. If a node in the network owns that resource, it responds with a service reply packet (RESOURCE REPLY) that is unicast back to the requester similar to the ROUTE REPLY packet of DSR. RESOURCE REQUEST packets contain sequence numbers to ensure that the overhead incurred is at most N (network size) packets. The remaining overhead is the number of RESOURCE REPLY packet transmissions, which is determined by the degree of replication of the resource being requested.

In Ekta-RD, the resource discovery application is built on top of the Ekta DHT substrate. Ekta provides three DHT APIs, *route(Message, Key), route(Message, IP Address,)* and *broadcast(Message, Broadcast Address)*, as well as an additional API *Proximity(IP Address)*. The *Proximity(IP Address)* interface returns the hop distance of the node specified by looking up the locally cached routes. If no such route is cached, Ekta returns null.

Ekta-RD simply relies on Ekta to route a RESOURCE REQUEST packet to the correct directory agent and receive a reply. The announcement of each available resource is inserted into Ekta via hashing the resource identifier into a *Key*, and invoking *route(Message, Key)* of Ekta with the *Message* containing the resource description. Similarly, when a resource is required, Ekta-RD hashes the resource identifier into a *key* and invokes *route(Message, key)* of Ekta. Because of the DHT abstraction, Ekta will route the announcement and queries for the same resource to the same node in Ekta—the directory agent for the mapped resource. The node replies to the requester with a list of nodes that own the resource in the network. The requester then finds the closest node out of this list using Ekta's *Proximity(IP Address)* interface and contacts the chosen node to use the resource. If Ekta cannot determine the proximity of any node in the list returned in the RESOURCE REPLY, the application randomly selects a node out of the list and contacts that node to use the resource. This will trigger a ROUTE REQUEST for that randomly chosen node by Ekta.

The experimental results[19] show that for the resource discovery application, the DHT-based approach Ekta-RD consistently outperforms the broadcast-based approach DSR-RD for a wide range of application parameters. Specifically, DSR-RD incurs comparable routing overhead to Ekta-RD for high inter-arrival time between resource queries. As the inter-arrival time decreases, Ekta-RD incurs up to an order of magnitude lower overhead compared to DSR-RD. Furthermore, these results hold true for a wide range of mobilities. These results suggest that efficient structured P2P substrates such as Ekta provide a viable and efficient approach to building distributed applications in mobile ad hoc networks.

47.6.3 Multicast

A wide variety of work has concentrated on developing overlay multicast protocols for the Internet,[5,11,26,52,70,72] due to the difficulties inherent in the router assisted IP multicast. Among these, many protocols specifically use structured P2P routing techniques for scalable multicasting.[10,11,52,70] Although no prior work has adapted structured P2P-based multicast protocols to MANETs, we discuss the issues associated with this approach using Scribe[11] as an example. Scribe is built on top of Pastry, which provides it with a generic object location and routing substrate. It leverages Pastry's reliability, self-organization, and locality properties to create and manage groups and to build efficient per-group multicast trees.

Ekta provides a structured object location and routing substrate for MANETs, similar to Pastry for the Internet, and thus can potentially be leveraged by Scribe to provide multicast services in MANETs. The reverse path tree built by Scribe/Ekta would be an overlay tree based on nodeIDs. Each node performs prefix-based routing to a root node (based on the hash of the group identifier). Consequently, many group members in a local region choose a nearby common node whose nodeID matches a longer prefix to the group identifier's hash than their own nodeID. That node, in turn, chooses another such node until the root node is reached and the tree is formed. Note that non-group members may also be chosen as part of the tree, depending on their nodeIDs and locations.

Although there has not been any study on P2P-based multicast in MANETs as described above, the performance benefit of this approach is potentially limited for the following reasons.

First, due to the constraints imposed by prefix-based routing in the namespace by Scribe/Ekta, the tree constructed for a group may not be as efficient as one constructed taking into account the physical connectivity among the nodes. MANET multicast protocols typically use some sort of broadcast mechanism to construct a multicast tree. For example, consider two non-leaf nodes (A and B) in a multicast tree that are within transmission range of each other. In Scribe/Ekta, leaf nodes can select either of these two nodes as parents, depending on their routing tables and nodeIDs. This can result in a multicast tree where both these nodes are internal nodes of the tree. However, in a broadcast-based approach to tree construction (e.g., Ref. 67), the nearby nodes would choose only one of these two nodes (say A) as a parent. This chosen node would then be an internal node of the tree. Note that in the broadcast case, one multi-hop route is needed to connect node A to its parent node. For Scribe/Ekta, both nodes A and B need to connect to their parent nodes via multi-hop routing, resulting in increased bandwidth consumption. Additionally, some tree building approaches for MANETs also take advantage of Wireless Multicast Advantage,[66] which further reduces their bandwidth costs by allowing multiple receivers to receive packets with one transmission.

Second, with node mobility, the overlay tree built by Scribe/Ekta must be continuously maintained and updated, which may cause increased overhead. Although low overhead prefix requests can be used to repair the overlay tree, Scribe/Ekta is likely to be less robust with increased mobility as compared to *mesh*-based MANET multicast protocols.[28,37] Mesh-based multicast protocols are better at dealing with link breaks due to mobility. Note that this trade-off is not specific to using a P2P overlay but is more generally about tree-based versus mesh-based schemes. Thus, to maintain a certain degree of robustness, Scribe/Ekta-based multicast is likely to pay a higher cost than previously proposed mesh-based MANET multicast protocols.

In general, most applications that involve all or a large subset of nodes in the MANET, such as broadcast and multicast, are likely to be more efficient if built directly on top of the physical broadcast mechanism compared to built on top of a structured P2P substrate. This is because an efficient structured P2P substrate will experience many flooding-based route discoveries in maintaining individual links in the overlay, while each such flooding can already reach all or a large number of nodes in the MANET.

47.7 Other Overlay-Based Approaches in MANETs

There have been several studies on building overlays in MANETs that involve a subset of nodes, that is, similar to the overlays in the Internet. So far, all of these studies have focused on constructing overlays to directly support multicast or unicast. That is, the overlay built in such studies are not meant to be general-purpose substrates for building distributed applications and services.

47.7.1 Multicast

Many multicast protocols have been proposed for MANETs[14,18,29,37,57] that maintain state at both group members and non-group members to support a multicast session. This potentially lowers the resilience of the protocols to node mobility due to the higher requirements of state maintenance. Subsequently, overlay multicast protocols for MANETs[23,69] have been proposed to improve robustness and reduce overhead at the cost of potentially sub-optimal multicast trees.

AMRoute[69] uses bi-directional unicast tunnels to organize the multicast group members into a virtual mesh. After mesh creation, one group member, designated the logical core, initiates the creation and maintenance of a shared data delivery multicast tree. To deal with node mobility, the tree is periodically rebuilt. AMRoute can be inefficient due to the use of a static virtual mesh in building a shared tree under network topology changes.

PAST-DM[23] was subsequently proposed to deal with these inefficiencies. PAST-DM continuously optimizes the quality of the multicast tree to reconcile the overlay and current physical topology. The idea is not use a static virtual mesh, but rather allow the virtual mesh to gradually adapt to changes in the underlying network topology. An adapted mesh can then result in better adapted multicast trees. The multicast trees in PAST-DM are locally constructed, source-based Steiner trees built using the Takahashi-Matsuyama heuristic.[63] PAST-DM also tries to eliminate redundant physical links to reduce the bandwidth cost of the multicast session.

A location guided tree[13] is a multicast tree constructed using location information (e.g., available from GPS) in a MANET, and thus is also a form of overlay multicast. In this approach, a source-based Steiner tree is constructed using the Takahashi-Matsuyama heuristic[63] for performing multicast. Unlike PAST-DM, geometric distances are used as link costs.

NICE-MAN[6] is an overlay-based multicast protocol for MANETs based on the NICE[5] application layer multicast for the Internet. NICE is a multi-source application layer multicast that creates a hierarchy of fully meshed clusters, that is, the sibling nodes sharing the same parent node are fully connected. Because of the full-meshed clusters, NICE is much more resilient to node failures than a simple-tree-based multicast protocol. To adapt NICE for MANETs, NICE-MAN makes three changes. First, a cluster leader continuously monitors the cluster members and transfers the cluster leader role to another cluster member if that cluster member becomes closer to the center of the cluster than itself. That is, the cluster leader is continuously reselected to maintain the parent-children locality of the overlay tree. Second, instead of RTT, the network distance metric is changed to use the number of hops in a multi-hop path provided by the underlying unicast routing protocol of MANETs (e.g., AODV). Third, the broadcast capability of the medium is exploited using the overlay NICE tree as a backbone and nodes within the transmission range of an overlay node do not explicitly join the NICE hierarchy.

47.7.2 Virtual Infrastructure

Many routing protocols developed, such as DSR and AODV, involve all nodes in route creation and maintenance while relying on broadcast relaying for route creation. Subsequently, many researchers have worked on routing solutions that enable protocols to use only a subset of the nodes and avoid broadcast relays. To this end, routing based on a connected dominating set (subset) was proposed, where the search space for a route is reduced to the nodes in the set. A set is defined as *dominating* if all the nodes in the system are either in the set or are neighbors (within the transmission range) of nodes in the set.

Constructing a connected dominating set effectively *overlays* a virtual infrastructure (core) in the MANET. Distributed and efficient computation of the connected dominating set has been studied extensively, and many routing protocols based on this approach have been proposed,[3,17,31,35,41,58–60,62,65,68]. Virtual infrastructures have been proposed for handling unicast, multicast, as well as QoS traffic.

Virtual infrastructures have also been proposed to provide various services in MANETs, such as mobility management[40] and service discovery.[36]

47.8 Summary and Future Research

In this chapter we presented a survey of support for peer-to-peer overlay abstractions in mobile ad hoc networks fostered by the recent advent of peer-to-peer overlay networks and systems in the Internet. We classify the existing approaches for supporting P2P overlay abstractions in MANETs according to the type of P2P overlays (structured versus unstructured) being modeled and the layer at which the P2P overlay protocol operates. In particular, if the protocol is layered over an existing routing protocol for MANETs, we classify it as the *layered design*; and if the protocol is integrated with a MANET multi-hop routing protocol, we classify it as the *integrated design*. Our classification excludes numerous works that build applications by directly exploiting the inherent peer-to-peer nature of ad hoc networks, for example, gossip-based data dissemination, or construct overlays to directly support multicast or unicast, for example, virtual infrastructures. Such studies, many of which were proposed before the advent of the Internet P2P overlays, do not leverage the P2P overlay abstraction.

The subject of exploiting peer-to-peer overlays in MANETs is relatively new. Many interesting problems require further research, including:

- Given the high dynamics in MANETs due to node mobility, which of the unstructured or structured overlay abstractions is more efficient in supporting common distributed applications such as file sharing?
- How can one efficiently integrate a DHT for the Internet with MANET routing protocols other than DSR to support the DHT abstraction in MANETs? For example, the use of hop-by-hop routing as in AODV[50] and DSDV[49] requires that all nodes along a route to a destination maintain this route. This implies that an integration with Pastry would require these nodes to have a prefix match with the destination nodeID in order to contain the destination nodeID in their prefix-based routing structures. With source routing, however, the intermediate physical hops need not maintain a route to the destination.
- How can one efficiently integrate other DHTs such as CAN and Chord with MANET routing protocols? A Chord-based DHT integrated with DSR would be similar to Ekta in that each node would store in its successor list and finger table a list of source routes to the corresponding nodeIDs. However, because the routing table entries for Chord are required to refer to specific points in the namespace, proximity-aware selection of overlay hops would be less flexible than in Pastry. This same problem would exist in a CAN-based integrated DHT.
- Can DHTs be leveraged to support scalable unicast in MANETs? Current unicast protocols for MANETs can support up to a few hundred nodes, while DHTs in the Internet have been shown to scale to millions of nodes.
- Can incentive techniques developed for P2P overlays in the Internet for encouraging peering nodes to cooperate be applied to MANETs?
- Can distributed security, trust, and reputation techniques developed in P2P overlays in the Internet be applied to MANETs?
- Can routing protocols developed in MANETs be leveraged to improve the performance and functionalities of P2P overlays in the Internet?

Acknowledgment

This research was supported by NSF grant ANI-0338856.

References

1. The Gnutella protocol specification, 2000. http://dss.clip2.com/GnutellaProtocol04.pdf.
2. The Gnutella 0.6 protocol draft, 2002. http://rfc-gnutella.sourceforge.net/.
3. K.M. Alzoubi, P.-J. Wan, and O. Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *Proc. of ACM MobiHoc,* June 2002.
4. D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proc. of ACM SOSP,* October 2001.
5. S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proc. of ACM SIGCOMM,* August 2002.
6. S. Blodt. Efficient end system multicast for mobile ad hoc networks. In *Proc. of the 1st International Workshop on Mobile Peer-to-Peer Computing (MP2P'04),* March 2004.

7. L. Breslau et al. Advances in network simulation. *IEEE Computer,* 33(5):59–67, May 2000.
8. J. Broch et al. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of ACM MobiCom,* October 1998.
9. M. Castro, P. Druschel, Y.C. Hu, and A. Rowstron. Exploiting network proximity in peer-to-peer overlay networks. Technical Report MSR-TR-2002-82, 2002.
10. M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: high-bandwidth multicast in a cooperative environment. In *Proc. of ACM SOSP,* October 2003.
11. M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. Scribe: a large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communication (JSAC),* 20(8), October 2002.
12. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P systems scalable. In *Proc. of ACM SIGCOMM,* September 2003.
13. K. Chen and K. Nahrstedt. Effective location-guided tree construction algorithms for small group multicast in MANET. In *Proc. of IEEE INFOCOM,* June 2002.
14. C. Chiang, M. Gerla, and L. Zhang. Forwarding Group Multicast Protocol (FGMP) for multihop, mobile wireless networks. *Baltzer Cluster Computing,* 1(2):187–196, 1998.
15. I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong. Freenet: a distributed anonymous information storage and retrieval system. In *Workshop on Design Issues in Anonymity and Unobservability,* July 2000. ICSI, Berkeley, CA.
16. F. Dabek, M.F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *Proc. of ACM SOSP,* October 2001.
17. B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Proc. of IEEE ICC,* Volume 1, pages 376–380, Montreal, Canada, June 8–12, 1997.
18. S.K. Das, B.S. Manoj, and C.S.R. Murthy. A Dynamic core based multicast routing protocol for ad hoc wireless networks. In *Proc. of ACM MobiHoc,* June 2002.
19. S.M. Das, H. Pucha, and Y.C. Hu. Ekta: an efficient peer-to-peer substrate for distributed applications in mobile ad hoc networks. Technical Report TR-ECE-04-04, Purdue University, January 2004.
20. S.R. Das, C.E. Perkins, and E.M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. of IEEE INFOCOM,* March 2000.
21. S. Deering, C. Partridge, and D. Waitzman. Distance vector multicast routing protocol. Technical Report 1075, IETF, November 1988.
22. FIPS 180-1. Secure Hash Standard. Technical Report Publication 180-1, Federal Information Processing Standard (FIPS), NIST, U.S. Department of Commerce, Washington, D.C., April 1995.
23. C. Gui and P. Mohapatra. Efficient overlay multicast for mobile ad hoc networks. In *Proc. of IEEE WCNC,* March 2003.
24. E. Guttman. Service location protocol: automatic discovery of IP network services. *IEEE Internet Computing,* 3(4):71–80, 1999.
25. C. Hendrick. Routing Information Protocol. RFC 1058, IETF, June 1988.
26. Y. hua Chu, S.G. Rao, and H. Zhang. A case for end system multicast. In *Proc. of ACM Sigmetrics,* pages 1–12, June 2000.
27. J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson. A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks, July 2001.
28. J.G. Jetcheva and D.B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *Proc. of ACM MobiHoc,* October 2001.
29. L. Ji and M.S. Corson. A lightweight adaptive multicast algorithm. In *Proc. of IEEE GLOBECOM,* November 1998.
30. D.B. Johnson and D.A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks.* Kluwer Academic, 1996.
31. J. Wu, M. Gao, and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. In *Proc. of ICPP,* September 2001.

Au: # 27 complete citation ? Ref. cited in text?

32. B. Karp and H. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proc. of ACM MobiCom,* August 2000.
33. KaZaA. Homepage http://www.kazaa.com.
34. A. Klemm, C. Lindemann, and O. Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *Proc. of IEEE VTC,* October 2003.
35. U.C. Kozat, G. Kondylis, B. Ryu, and M.K. Marina. Virtual dynamic backbone for mobile ad hoc networks. In *Proc. of IEEE ICC,* June 2001.
36. U.C. Kozat and L. Tassiulas. Network layer support for service discovery in mobile ad hoc networks. In *Proc. of IEEE INFOCOM,* April 2003.
37. S.-J. Lee, W. Su, and M. Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. *Kluwer Mobile Networks and Applications,* 2000.
38. J. Li, J. Jannotti, D.S.J.D. Couto, D.R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proc. of ACM MobiCom,* August 2000.
39. Z. Li, B. Li, D. Xu, and X. Zhou. iFlow: middleware-assisted rendezvous-based information access for mobile ad hoc applications. In *Proc. of MobiSys,* 2003.
40. B. Liang and Z.J. Haas. Virtual backbone generation and maintenance in ad hoc network mobility management. In *Proc. of IEEE INFOCOM,* March 2000.
41. C.R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications,* 15(7):1265–1275, 1997.
42. I.F. Matei Ripeanu. Mapping the Gnutella network macroscopic properties of large-scale peer-to-peer systems. In *Proc. of IPTPS,* 2002.
43. M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks, 2001. **Au:** # 43-incomplete citation
44. J. Moy. Multicast extensions to OSPF. Technical Report 1584, IETF, March 1994.
45. J. Moy. OSPF version 2. RFC 1583, IETF, March 1994.
46. J.C. Navas and T. Imielinski. Geographic addressing and routing. In *Proc. of ACM MobiCom,* 1997.
47. L.B. Oliveira, I.G. Siqueira, and A.A. Loureiro. Evaluation of ad-hoc routing protocols under a peer-to-peer application. In *Proc. of IEEE WCNC,* March 2003.
48. M. Papadopouli and H. Schulzrinne. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *Proc. of ACM MobiHoc,* October 2001.
49. C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. of ACM SIGCOMM,* August 1994.
50. C.E. Perkins and E.M. Royer. Ad hoc on-demand distance vector routing. In *Proc. of the IEEE WMCSA,* February 1999.
51. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. schenker. A scalable content-addressable network. In *Proc. of ACM SIGCOMM,* August 2001.
52. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using content-addressable networks. In *Proc. of the Third International Workshop on Networked Group Communication,* November 2001.
53. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: a geographic hash table for data-centric storage in sensornets. In *Proc. of 1st ACM WSNA,* September 2002.
54. Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, IETF, March 1995.
55. A. Rowstron and P. Druschel. PAST: a large-scale, persistent peer-to-peer storage utility. In *Proc. of ACM SOSP,* October 2001.
56. A. Rowstron and P. Druschel. Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of Middleware,* November 2001.
57. R. Royer and C. Perkins. Multicast using ad-hoc on demand distance vector routing. In *Proc. of ACM MobiCom,* August 1999.
58. P. Sinha, R. Sivakumar, and V. Bharghavan. Enhancing ad hoc routing with dynamic virtual infrastructures. In *Proc. of IEEE INFOCOM,* Anchorage, AK, April 2001.

59. R. Sivakumar, B. Das, and V. Bharghavan. Spine routing in ad hoc networks. *ACM/Baltzer Publications Cluster Computing Journal, Special Issue on Mobile Computing,* 1(2):237–248, 1998.
60. R. Sivakumar, P. Sinha, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications,* 17(8):1454–1465, August 1999.
61. I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup service for internet applications. In *Proc. of ACM SIGCOMM,* August 2001.
62. I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Trans. Parallel Distrib. Syst.,* 13(1):14–25, 2002.
63. H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Mathematica Japonica,* 24:573–577, 1980.
64. C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *Proc. of ACM SIGCOMM,* August 2004.
65. P.-J. Wan, K.M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Netw. Appl.,* 9(2):141–149, 2004.
66. J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *INFOCOM 2000,* Tel-Aviv, Israel, March 26–30, 2000.
67. J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. Energy-efficient broadcast and multicast trees in wireless networks. *Mob. Netw. Appl.,* 7(6):481–492, 2002.
68. J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Wireless Networks in the Telecommunication Systems,* 3:63–84, 2001.
69. J. Xie, R.R. Talpade, A. Mcauley, and M. Liu. AMRoute: ad hoc multicast routing protocol. *Mob. Netw. Appl.,* 7(6):429–439, 2002.
70. R. Zhang and Y.C. Hu. Borg: a hybrid protocol for scalable application-level multicast in peer-to-peer systems. In *Proc. of NOSSDAV,* June 2003.
71. B.Y. Zhao, J.D. Kubiatowicz, and A.D. Joseph. Tapestry: an infrastructure for fault-resilient wide-area location and routing. Technical Report UCB//CSD-01-1141, U.C. Berkeley, April 2001.
72. S.Q. Zhuang, B.Y. Zhao, A.D. Joseph, R.H. Katz, and J. Kubiatowicz. Bayeux: an architecture for scalable and fault-tolerant wide-area data dissemination. In *Proc. of NOSSDAV,* June 2001.