APSALAR: Ad hoc Protocol for Service-Aligned Location Aware Routing

Warren Kenny
Distributed Systems Group
Department of Computer Science
Trinity College, Dublin, Ireland
kennyw@cs.tcd.ie

Stefan Weber
Distributed Systems Group
Department of Computer Science
Trinity College, Dublin, Ireland
sweber@cs.tcd.ie

ABSTRACT

Current solutions to communication in mobile ad hoc networks (MANETs) are based on the use of IP addresses. These approaches identify the nodes involved in data transmission through the same type of address used in traditional wired networks.

However, IP addresses do not carry the same meaning in MANETs as in wired networks, where routing protocols can reduce the routing effort based on the assumption that stub networks are physically static. In MANETs, where currently similar routing protocols are used to determine routes to individual nodes, IP addresses as a basis for routing decisions represent a poor choice, because nodes in these networks are assumed to be highly mobile.

We argue that a structured Peer-to-Peer (P2P) protocol that is aware of the proximity between nodes represents a viable alternative to IP in MANETs. Our approach replaces IP ad- dresses in favour of a service-oriented structured P2P identifiers that exploits a node's proximity-awareness in order to form physically-close clusters. In this paper, we describe the design of our routing protocol, called APSALAR, and discuss its features in comparison to existing approaches.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communications—*Peer-to-Peer, MANETs*

General Terms

Service-Aligned Routing

Keywords

Peer-to-Peer, MANETs, location-awareness, service-oriented computing

CAMS 2009, June 16, Dublin, Ireland

1. INTRODUCTION

IP addresses do not carry the same meaning in MANETs as in wired networks. Routing protocols in wired networks reduce the routing effort based on the assumption that a stub network with its address range represents a grouping of nodes and that these stub networks are physically static. Current approaches to routing in MANETs employ similar routing protocols based on IP addresses in order to determine routes to individual nodes. However, the individual nodes that take the place of stub networks in traditional routing protocols are assumed to be highly mobile in MANETs. This means that the underlying assumption of physically-static stub networks that represent a range of IP addresses does not hold anymore and that IP addresses only function as unique identifiers without added co-notations that are exploited for routing.

However, the use of IP addresses solely as unique identifier in MANETs is questionable. An application generally connects to a specific node in order to communicate with a service on this node. In wired networks, this is facilitated through service brokers that return the identity of a node that hosts a requested service. An application that requests a service, is interested in the provision of the service, not necessarily a specific host; thus the indirection through the resolution of unique identifier raises the question if a direct resolution of the service would be more appropriate.

We argue that a structured Peer-to-Peer (P2P) protocol that is aware of the proximity between nodes represents a viable alternative to IP in MANETs. Our approach replaces IP addresses in favour of service-oriented identifier that exploit proximity-awareness in order to form physically-close clusters.

P2P overlays are designed to provide decentralized mechanisms to transfer data between peers. Compared to unstructured P2P overlays, which resolve searches for content through broadcasting requests, structured P2P overlays associate a node's overlay identifier with the content held by that node. By having each node in an overlay form connections only with its neighbours in the ID space, requests for content can be routed towards their destination without flooding the network or maintaining a centralized indexing authority. This overlay design is referred to as a Distributed Hash Table (DHT). Current implementations of DHT in infrastructure networks include Pastry [7] and Chord [8]. These approaches allow requests for content to be routed in Olog(N) hops in networks of N nodes without resorting to request flooding.

A number of projects have investigated the application of

structured P2P overlays to MANETs; Virtual Ring Routing [1] (VRR) is an implementation of DHT on top of MANETs with several added features for increasing efficiency and reliability. One issue with VRR is that the mobility of MANETs introduces inherent inefficiency into the routing process, as neighbours in overlay-space may be many hops apart in physical space. This means that in order to maintain the integrity of the overlay, nodes must maintain links with overlay neighbours which are physically distant in the network, as shown in figure 1. MADPastry [9] and PeerNet [3] address this problem by allowing node identifiers to change depending on their physical location.

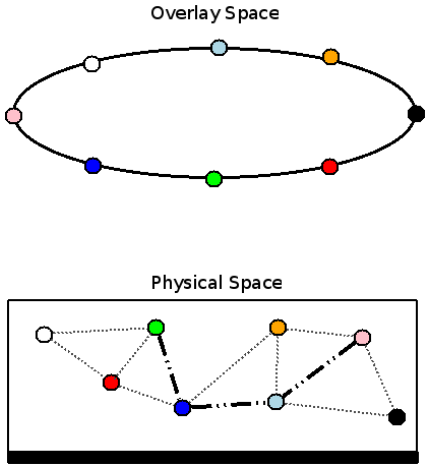

Figure 1: Disparity Between Node Connections in Physical and Overlay Space

In comparison to MADPastry and PeerNet, APSALAR exploits knowledge about the proximity of neighbouring nodes to form physically-close clusters. This knowledge and the services that a node provides are integrated in the identifiers that a node distributes amongst its neighbours, resulting in a hybrid protocol that combines advantages from both reactive and proactive protocols in order to derive a scalable protocol.

In the following section, we will discuss the related work in the area of unstructured and structured routing protocols for MANETs. This will be followed by a description of our approach and a description of the overall system architecture. Then we will discuss the approach in comparison to existing approaches and present the conclusions can be drawn from this comparison.

2. RELATED WORK

In this section we will give examples of unstructured and structured routing protocols, demonstrating the advantages that structured protocols possess when locating content. We will also illustrate the key differences between the location-aware structured overlays *PeerNet* and *MADPastry*.

2.1 AODV

AODV is a *reactive* routing protocol, meaning that route discovery happens on request. When a node in a MANET running AODV attempts to make a connection with a specific node, it broadcasts a *route request* to all of its neighbours. Assuming that one of the source node's neighbours isn't the destination, each neighbour re-broadcasts the request to its neighbours which repeat the process until the route request reaches the destination node. The destination's reply is routed back along the nodes which relayed the request to the source, which chooses the route that took the fewest hops to reach its destination. This process of broadcasting and re-broadcasting requests is known as *flooding* and it contributes significantly to the amount of network traffic in an AODV-based MANET. Unless the route request relay process is limited using a TTL value, the amount of traffic generated by route requests in a large, mobile network may exceed the amount of data traffic in the network.

The unstructured P2P protocol Gnutella [6] uses a similar system for locating content in a network. Each node in a Gnutella network maintains a neighbour list to which it broadcasts content location requests. Each of these neighbours sends the content request to its own neighbours until the content is found, at which point the source is informed as to the address of the content holder and the two nodes connect to each-other. As shown in [6], this protocol has major scalability issues due to the amount of traffic generated by content location requests.

2.2 VRR

Virtual Ring Routing (VRR) [1] is a routing protocol for MANETs based on distributed hash tables. VRR maintains links between ID-space neighbours by routing packets through physical neighbours. Each VRR node maintains routing information not only for its ID-space neighbours but also for all routes for which the node acts as an intermediate hop. Thus, when a node receives a packet bound for a node with a certain ID, it will forward that packet along the path whose endpoint identifier is numerically closest to that packet's destination. This feature decreases the number of hops that need to be traversed for a packet to reach its destination; rather than strictly following the route described by the overlay ring. It also increases reliability, as there is probably an alternative path to a destination node when an existing path fails.

VRR introduces several methods for repairing paths and neighbour lists in the event of node failure as well as methods for merging network partitions. Evaluation of VRR demonstrated that for an increasing number of constant bit- rate flows over various areas with variable node densities, the system out-performed unstructured routing protocols both in terms of generated overheads and delivery reliability.

2.3 MADPastry

MADPastry [9] is a DHT implementation on top of MANETs which combines principles of Landmark Routing with *location-prefixes* in overlay IDs. A MADPastry-based network is divided into a series of clusters, with each node in a cluster having a certain location prefix in their overlay ID. One node in each cluster, the *landmark node*, broadcasts this location-prefix to nearby nodes which then adopt it as part of their identifiers. When a node migrates between clusters, it adopts the location prefix of the destination cluster and appends a new random identifier; thus the identity of a node doesn't remain intact as it transitions from one cluster to another.

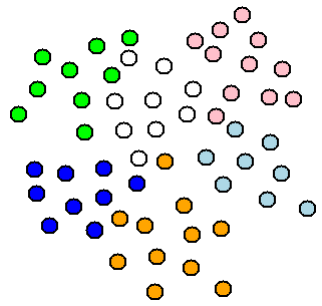

Figure 2: The Clustering Tendencies of MADPastry Nodes (shade denotes location prefix)

This results in an overlay topology which more closely matches that of the physical network topology as nodes with similar ID prefixes will tend to be close in ID-space, thus reducing the number of hops between overlay neighbours. Figure 2 illustrates the clustering tendency of MADPastry. Routing within clusters and between the outer edges of clusters is accomplished using distance-vector routing or simple broadcasting when no route to the destination is available in a node's routing table.

While MADPastry has been shown to improve upon location-unaware DHT overlays and unstructured protocols, disparities between the overlay and the physical topologies may still form, especially at the inter-cluster level. Clusters which are nearby in physical space may be far apart in overlay space, and vice-versa; thus decreasing efficiency when communicating between clusters. Additionally, the fact that MADPastry nodes do not maintain any part of their ID when they transition between clusters makes maintaining connections with such nodes more difficult.

2.4 PeerNet

PeerNet [3] also adopts the use of location-specific address prefixes for resolving differences between the physical and overlay topologies. Unlike VRR and MADPastry, however, PeerNet forms a binary tree structure in overlay space, with each level of the binary tree having a specific prefix. This tree structure maintains routing efficiency while simplifying the process of joining and leaving the network. However, PeerNet maintains separation between a node's address and its overlay identifier. In PeerNet, a node's identifier is static and unique while its overlay address changes depending on its location in the physical network. In order to route to a node with a specific identifier, a PeerNet node must look up that node's current overlay address using the PeerNet distributed lookup service.

This lookup service is based on the correlation between PeerNet nodes' addresses and identifiers; the node that holds information about the target node's current address is the node with the minimum *XOR distance* between its address and target identifier. PeerNet preserves lookup locality by distributing updates regarding node address changes throughout the network. The methods used for looking up addresses makes it likely that a node near to the source of the request will have the requested information, thus ensuring that an address request need not traverse the entire network. Once a node's address has been found, messages can be routed to that node by navigating the binary tree, comparing level-prefixes at each hop until the appropriate level and node have been reached.

PeerNet represents an interesting approach to resolving changes in physical network topology by using an overlay topology and its node lookup service is efficient. However, it separates a node's identifier from its address. This decision makes its design unsuitable for service- oriented communication.

3. OUR APPROACH

APSALAR was designed with two goals in mind: 1) The communication among nodes should exploit the fact that a wireless transmission is a localized broadcast and 2) the identification of the destination is in form of a service rather than the identity of a specific node. These goals are addressed by a protocol that employs location prefixes similar to MADPastry, an ID update mechanism similar to PeerNet and a service-oriented routing protocol.

APSALAR's primary contribution is to combine well-known and novel techniques in MANET cluster formation, routing and DHT-based lookups, resulting in a protocol which optimizes routing in MANETs for service-oriented computing. Additionally, we strive to accomplish these goals while minimizing resulting network overheads by reducing inter-cluster communications and the use of flooding.

In the following, we will first describe our approach to the formation of clusters, followed by an explanation of intra- and inter-cluster communication, a description of inter-cluster mobility and the representation of services in our approach.

3.1 Proximity-Aware Clustering

Each node in APSALAR describes its services through an ID that consists of 3 parts: A cluster identifier, a service prefix and a unique node ID. A cluster identifier associates a node as belonging to a cluster of physically-close nodes, a set of service bounds express the services that a node offers and a unique node ID distinguishes an individual node from all nodes in the network.

A cluster is formed by employing locally-limited cluster identifiers. These cluster identifiers are associated with pre-designated nucleus keys. A node that joins a network and listens for HELLO packets from neighbouring nodes. These HELLO packets are broadcast periodically and contain cluster identifiers as well as information about neighbouring nodes of the node that broadcasts the HELLO packet. If the cluster identifiers of neighbouring nodes are closer to the nucleus key of a node, it adopts the cluster identifier of its neighbours; otherwise it derives its cluster identifier from its nucleus key.

A node begins broadcasting a cluster identifier, if it has established that it is itself a nucleus node. Nodes will designate themselves as nucleus nodes is no nucleus node broadcasts are detected. When several nucleus nodes meet, the nuclei with the fewest child or *cadre* nodes attached to them will adopt the cluster identifier of the one with the most cadre nodes and cause their own cadre to become a part of the largest. At this point all nodes that are within the predetermined hop radius of the nucleus node will have adopted its cluster identifier into their addresses.

The broadcasts of these cluster identifiers are associated with a time-to-live (TTL). Nodes that receive these broadcasts decrease the TTL and re-broadcast cluster identifier if

the TTL is not zero. This TTL allows the control of the size of clusters and a node with a TTL of zero knows that it is at the edge of a cluster.

Once a node has adopted a cluster identifier, it sends a message around the DHT ring in order to determined the addresses of all nodes within the cluster. Once cluster membership data has returned to the joining node, it determines which nodes should act as its ID-space neighbours based on their ID and forms connections with them in order to join the DHT ring.

Where APSALAR's design diverges from that of MADPastry is its approach to overlay ring formation. Rather than having a DHT overlay spanning the entire network, DHT rings are formed inside each cluster only, with inter-cluster communication accomplished through the direct routing of messages to boundary nodes. Thus, the cluster identifier has no connection with the formation of the local DHT except in ensuring the locality of formed connections. Figure 3 illustrates the ID-space clustering pattern that results from the above approach.

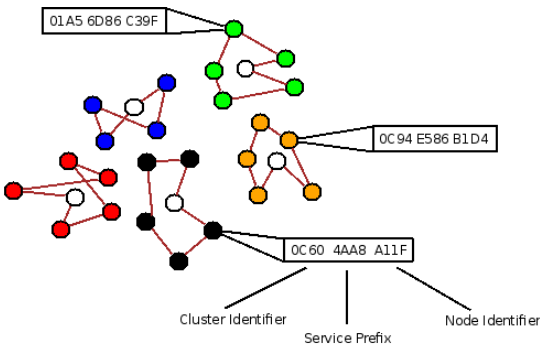

Figure 3: Clustering Pattern

The result of this clustering technique is similar to that of MADPastry: Nodes which are physically close to each-other are also close in ID-space. This results in greatly reduced overhead from routing as the number of hops between ID-space neighbours drops. Our approach also improves upon MADPastry's approach in that nodes which are on the boundary of a cluster do not have to maintain ID-space links with nodes which are physically distant in order to maintain the integrity of a network-wide DHT.

3.2 Intra-Cluster Communication

APSALAR's intra-cluster routing protocol closely resembles that of VRR. In an APSALAR cluster, each node maintains a virtual neighbour table and a physical neighbour table. These tables are filled by listening for the HELLO packets that are periodically broadcast by all APSALAR nodes.

APSALAR improves upon VRR by including additional information in its HELLO messages. Specifically, APSALAR nodes will broadcast not only their own identifiers to their neighbours, but also a list of all of their physical neighbours. This allows nodes to route packets more efficiently as it means that, even though one of a node's physical neighbours may not have an identifier which is closer to a packet's destination than that of its other neighbours, that same node may have neighbours which are closer or even include the destination; thus making it the most suitable intermediate hop along the route. Figure 4 illustrates the neighbour list dissemination process.

Figure 4: Neighbour Lists are Broadcast by Cadre Nodes

APSALAR nodes will always attempt to forward a packet to the neighbour or neighbour's neighbour whose identifier is closest to that packet's destination. As already illustrated by VRR, this simple approach to routing produces high delivery success rates without the need for flooding.

3.3 Inter-Cluster Communication

Certain nodes on the edges of APSALAR clusters may be in contact with adjacent clusters through other edge nodes. If a HELO packet is intercepted which indicates that the source node has a different location prefix from that of the listener, the listening node is on the boundary between two clusters of APSALAR nodes and can route packets between them.

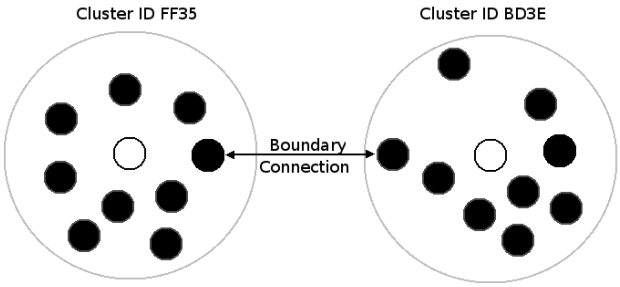

Figure 5: Nodes at Cluster Boundaries Form Links

Communication between clusters primarily occurs when the local cluster does not contain a required service. In this case, an attempt is made to communicate with adjacent clusters in an effort to find the required service. Packets with destination prefixes comprised of all-ones indicate that the destination is not in the local cluster and the packet should be sent to all nearby clusters. When a boundary node receives this packet, it sends it to its counterpart in the adjacent cluster. That node is then responsible for finding a server in the local cluster which can provide the required service. The destination cluster identifier of the packet is reset to that of the local cluster and an attempt is made

to send the packet to a node within the cluster that runs the needed service. See the section above on intra-cluster communications for more details.

3.4 Service-Aligned Routing

In order to optimize our system for service-oriented computing, we adopt the idea of *service prefixes*. A node's service prefix can be used to passively advertise the services it provides to the network. Thus each node's overlay ID is split into 3 distinct parts; the cluster identifier, service prefix and unique ID. This means that nodes which are physically close are also close in ID-space and that nodes which provide certain services can be found once the prefix denoting that service is known. This also means that APSALAR can route requests to the nearest node in the network which provides a specific service and that it can route to specific identifiers.

A service prefix is made up of two parts; a lower service bound and an upper service bound. These bounds describe the range within which all of a nodes advertised service identifiers fall. This approach is a compromise between advertising single service identifiers and advertising every service identifier as part of addresses. Using a bound-description approach means that service requests can be routed to the portion of a cluster containing nodes whose service bounds encapsulate the identifier of the needed service. These nodes will then evaluate whether they are running the requested service and either respond positively or pass the request on to the next node in the ring with an appropriate service prefix.

4. SYSTEM ARCHITECTURE

APSALAR is being implemented using OPNET; a network protocol design and simulation tool. Currently development of the socket interface and transport layer is in progress. Figure 6 shows the architecture diagram for APSALAR. The following subsections will elaborate on the role and design of each layer in the APSALAR protocol.

APSALAR has been designed specifically to work in wireless networks based on 802.1x technologies as a replacement for TCP/IP.

4.1 Application-Layer Compatibility

APSALAR has been designed to be compatible with existing applications that use sockets for communication. This is accomplished using the correlation between the sought service and end-point TCP or UDP socket for a connection; the end-point port number thus acts as a service identifier. The destination IP address specified by the socket's connect function is ignored unless an attempt is being made to send the connection request packet through the configured gateway, in which case the destination IP is encapsulated in a service connection packet routed towards a node serving as a gateway for the cluster. Otherwise, an attempt is made to route the connection request to the nearest node that provides the needed service.

4.2 Transport Layer

The Transport Layer is responsible for ensuring that packets are delivered reliably and in order. Our current design integrates principles of Explicit Link Failure Notification [4], a modification of TCP that integrates notifications from the underlying routing protocol regarding link failures. Holland et al concluded that the performance of TCP degrades significantly a node mobility increases, due in part to TCP's inability to recognise the difference between link failure and congestion. ELFN has been shown to degrade throughput by an average of 5% in static ad-hoc networks but increase performance by up to a factor of 7 in mobile networks [5].

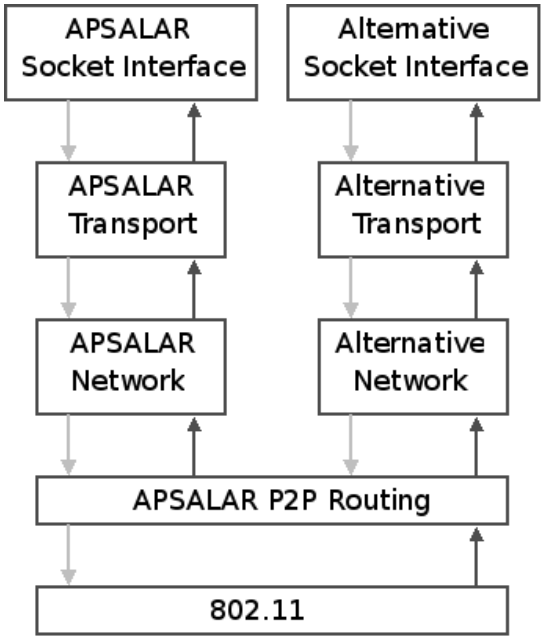

Figure 6: APSALAR Architecture

Utilization of ELFN as a basis for our transport layer greatly increases APSALAR's performance in situations where nodes are highly mobile. A new field is added to the ELFN header during encapsulation which specifies the node's unique ID. This field is used by destination nodes to ensure that reply packets reach the source node.

4.3 Network Layer

Data packets encapsulated by the transport layer are subsequently encapsulated with APSALAR address information by the network layer. Socket calls such as connect() which are translated by the socket interface specify a target service identifier. The APSALAR network layer converts this information into a destination address for the outgoing data and this is prepended to the packet before being sent to the routing layer. Connection establishment tends to involve service discovery, and thus the unique ID portion of the APSALAR address is set to FFFF, indicating that the source is searching for any node that runs the required service.

Data packets sent by socket calls such as send() result in the network layer specifying an APSALAR address with a specific unique ID; generally the unique ID associated with the server node which replied to the initial service discovery and connection request.

4.4 Routing Layer

The APSALAR routing layer is responsible for routing packets to their destination and determining which cluster a node should join. The routing layer interprets cluster data packets sent out by nucleus and cadre nodes, noting the network layer addresses specified and re-broadcasting as necessary. Network layer addresses are interpreted as being simple integers, regardless of the specific format used; thus allowing the APSALAR routing layer to be utilized by any network layer protocol.

The routing layer uses a hybrid reactive/proactive protocol to route packets to its neighbours in ID-space. Broadcasts from neighbouring cadre nodes regarding their neighbour lists are cached and, if the destination node for a given transmission is contained in one of these neighbour lists, the neighbour adjacent to the desired destination is as a relay. The routing layer attempts to maintain a connection with neighbours in overlay-space using this hybrid approach; this is how the DHT ring is formed inside each cluster.

5. DISCUSSION

APSALAR has several improvements over protocols such as VRR, MADPastry and PeerNet. Our approach presents a compromise between DHT integrity and network overhead generation by abandoning the whole-network DHT formation espoused by MADPastry in favour of a clustered approach. We believe that this approach is an improvement in that it avoids the difficulty of maintaining links between distant clusters which are adjacent in ID-space while still allowing DHT-based routing to function. In the case where a service cannot be found in the local cluster, APSALAR routes requests to all nearby clusters; thus allowing connections to be formed with the nearest provider of a given service without significantly increasing network overheads.

Implementation of ELFN in the APSALAR transport layer is also expected to result in greater performance due to its ability to adapt to mobility and link failures more effectively than TCP. We believe that the combination of a hybrid structured routing protocol, node clustering and an improved transport layer will allow APSALAR to outperform service-oriented approaches based on both structured and unstructured routing protocols.

6. CONCLUSIONS

As MANETs become more popular, we believe that service-oriented computing will define the standard way of using them. We have presented APSALAR as an efficient foundation for the formation and utilization of such networks. Our system utilizes the latest advancements in P2P protocol research to form networks which change configuration based on their physical topology in order to achieve high routing efficiency. Our clustering and routing approach means that service requests can always be efficiently routed to the nearest node which provides that service without the need for excessive flooding.

Our future work will include a full evaluation of APSALAR against existing structured and unstructured routing protocols in simulation as well as an actual implementation of the protocol outside of the OPNET simulation environment.

7. REFERENCES

[1] M. Caesar, M. Castro, E. B. Nightingale, G. Oï£¡Shea, and A. Rowstron. Virtual ring routing network routing inspired by dhts. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'06)*, pages 351–362, Pisa, Italy, September 2006.

[2] P. E. Engelstad and Y. Zheng. Evaluation of service discovery architectures for mobile ad hoc networks. *wons*, 0:2–15, 2005.

[3] J. Eriksson, M. Faloutsos, and S. Krishnamurthy. Peernet pushing peer-to-peer down the stack. In *Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, pages 268–277, Berkeley, CA, USA, February 2003.

[4] G. Holland and N. H. Vaidya. Analysis of tcp performance over mobile ad hoc networks. pages 219–230, August 1999.

[5] P. S. J.P. Monks and V. Bharghavan. Limitations of tcp-elfn for ad hoc networks. October 2000.

[6] M. Ripeanu. [15] peer-to-peer architecture case study: Gnutella network. *p2p*, 00:0099, 2001.

[7] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'01)*, pages 329–350, Heidelberg, Germany, November 2001.

[8] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, F. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, February 2003.

[9] T. Zahn and J. Schiller. Madpastry a dht substrate for practicably sized manets. In *Proceedings of 5th Workshop on Applications and Services in Wireless Networks (ASWN'05)*, Paris, France, June 2005.