Service Discovery in Ad hoc networks

Joakim Flathagen

25th June 2007

Contents

1 Introduction

Mobile computing is now a common part of our everyday life. We are moving away from stationary devices to ubiquitous devices. Mobile users can now access a myriad of information whenever and wherever needed. As an example, a tourist visiting a foreign city can easily access the Internet, check email, browse for the closest and cheapest restaurant and download electronic maps using a cheap handheld device. The change-over to ubiquitous computing is happening thanks to all the research done in the terms of microelectronics, mobile computing and mobile ad-hoc networks during the last decades.

A mobile ad-hoc network is a collection of mobile nodes connected by wireless link able to dynamically form an autonomous multihop radio network without the use of any preexisting infrastructure. Such a network is most commonly known as a MANET. The most important features of this infrastructure-less, dynamic and heterogeneous collection of MANET nodes is that it is self forming, autoconfigurable and adaptive to rapid changes of the network topology. Intermediate nodes in a MANET can act as routers that forward packets on behalf of other nodes.

Although early ad-hoc research was aimed to military networks, MANET technology is now attractive to a wide area of applications. Search and rescue operations, vehicle to vehicle networks, tactical networks, virtual classrooms, entertainment, sensor networks are all areas where great benefit can be achieved using the flexibility, ease of maintenance, auto-configuration, and the cost advantages of MANETs.

The discovery of services and resources is a crucial feature and a major component for the usability of mobile ad-hoc networks. Service discovery protocols let devices discover and take use of shared resources in the network, and further advertise their own capabilities to other devices. Examples of services shared in a MANET can be printers, databases, web servers, Internet gateways, sensors, weather forecast service, chemical and biological alarms, presence reports, different subscriptions etc.

In traditional networks and in the Internet community a lot of work has been done in the terms of service discovery. A number of mechanisms are proposed, such as Service Location Protocol (SLP), Jini, UPnP among others. However, fundamental differences in terms of computing resources, network bandwidth, and issues such as mobility and stability in the network environment makes these generic service discovery techniques unsuitable for ad-hoc networks. Such networks requires specialized solutions, which will be discussed in this document.

2 Background

This section addresses some of the MANET research topics, including routing and service discovery.

2.1 Ad hoc networks

Many applications can benefit from ad-hoc technology. Nevertheless, the flexibility, cost benefit, ease of maintenance, auto-configuration and all the other advantages come at a price. Ad hoc networking introduces a great many challenges and imperatives, and adopts the side effects of wireless computing: Wireless links is significantly less reliable than wired media, it has unpredictable signal quality and transmission range, the channel can be time-varying and possible asymmetric, and the wireless link suffer from security problems not found in wired networks. Further, the multhop nature in MANETs introduces a lot of other problems due to the topology dynamics, heterogeneity, variations of node availability, power constrains and so on.

A lot of research is done to improve the performance of ad hoc networks. MAC protocol performance, physical layer radio technology, routing protocol and service availability is some of the areas which affect the perception of a well built MANET. The following subsections will give an introduction to subjects regarding the two latter areas, namely ad-hoc routing and service discovery.

2.2 Ad hoc routing

Traditional routing protocols designed for fixed networks are in general not suited for the ad hoc environment. The dynamic topology, limited bandwidth, power constraints etc in MANETs requires tailor made solutions.

It is worth mention the fact that even if they are named “routing protocols”, the protocol does not in fact *route* packets. The responsibility of the protocol is to obtain topology information and to create routing tables for the operating system to use.

In the emerge of mobile ad-hoc networks, mainly two different routing approaches is considered: Reactive routing and proactive routing. Reactive routing protocols only attempts to discover routes between nodes on-demand. Proactive routing protocols on the other hand, seek to maintain routes to all nodes regardless of upper layer communication demands. A variety of protocols is proposed in both categories. Some of the routing protocols are considered purely academic, while others are heavily deployed in real-life applications.

The IETF is chartered to develop one reactive, and one proactive routing protocol through the MANET working group. The work has concluded in two RFC standards, namely AODV[15] and OLSR[5].

2.2.1 Reactive routing - AODV

Using the reactive approach, one can lower the total overhead using the protocol on cost of the initial delay finding the optimal route. Protocols in this category are also named source initiated or on-demand routing protocols. Some examples of these reactive routing protocols are AODV[15], DSR, TORA and DYMO[1]. In this section, AODV will be explained, as it is the most studied reactive protocol.

The Ad hoc on-demand distance vector protocol (AODV)[15] is representative for reactive routing. The protocol aims to obtain routes on-demand, i.e when an upper layer communication packet is destined to a node not known in the routing table. AODV uses three control messages to obtain and maintain routes; route request (RREQ), route reply (RREP) and route error (RERR).

The source broadcasts route request (RREQ) to the MANET if the routing entry is empty for the given destination. A node replies to this RREQ by sending a RREP message either if (i) it is the destination, or (ii) if it is an intermediate node and has a fresh route to the destination. If the destination is not known, the intermediate node will rebroadcast the RREQ. When a node re-broadcasts a Route Request, it sets up a reverse path pointing toward the source. This reverse path is used to forward Route Reply (RREP) unicast back to the source.

If a node is unable to forward packet, it generates a RERR message. When the originator node receives the RERR, it initiates a new route discovery for the given route. To avoid broadcast storm, AODV uses expanding ring search technique with increasing TTL for each unsuccessful attempt to obtain a route.

2.2.2 Proactive routing - OLSR

Using periodically exchanged control messages, the proactive routing system can provide immediate routes. Some examples of proactive (or table-driven) routing protocols are FSR, FSLS, OLSR[5], TBRPF and WOSPF. OLSR is probably the most used proactive routing protocol, and is further explained in the following section.

The Optimized Link State Routing Protocol (OLSR)[5] for MANET is a proactive, link-state routing protocol where each node maintains topology information by periodically exchanging link-state messages. The novelty of OLSR is to employ multipoint relays (MPR) to minimize the number of control messages flooding in the network. MPRs of a node is a subset of its one-hop neighbors chosen in a way such as the MPRs will cover all two-hop away neighbors.

Core functioning of OLSR is: (i) Packet format and forwarding; (ii) Link sensing with hello messages; (iii) Neighbor detection; (iv) MPR selection and MPR signaling; (v) Topology control message diffusion; (vi) Route table computation; (vii) Node configuration. These functions are made possible using the main OLSR messages such as HELLO, Topology Control (TC), and Multiple Interface Declaration (MID).

HELLO messages are exchanged between neighbors only, and diffuses information about the one-hop neighbors of a node. Upon reception of HELLO messages, the two hop neighborhood can be discovered, and further the MPRs of the given node can be chosen. The MPRs chosen by a node is further marked in the following HELLO messages broadcasted by that node.

In OLSR, all nodes chosen as MPR will transmit TC messages. The TC messages contains the address of the node generating the message, as well as the list of nodes that has chosen the given node as MPR (MPR selectors). TC messages are further flooded using the MPRs, disseminating network topology information to all the nodes in the OLSR network.

The MID message is broadcasted by nodes running OLSR on more than one network interface. A fourth message type is called Host and Network Association (HNA) message. The HNA message disseminates information about OLSR nodes that act as gateways (etiher to the Internet or to a separate ethernet) by using MPR flooding.

2.2.3 Way forward

The IETF MANET WG is now working on the development of the successors to AODV and OLSR, namely DYMO[1] and OLSRv2[4]. DYMO is built upon AODV's distance vector routing. OLSRv2 evolves from OLSRv1 and improves OLSR with increased extensibility, more efficient messaging and lower complexity.

The new MANET packet and message format PacketBB[3], is a part of both DYMO and OLSRv2. The PacketBB is a general purpose message format for information exchange between MANET routers. This message format makes it possible to add functionality to the routing protocol, and to create cross layer protocols such as IP-autoconfiguration, power-aware add-on and of course service discovery protocols as a part of the routing architecture.

3 Service Discovery

3.1 Introduction

A minimum user intervention and high flexibility is essential in ad hoc networking. A key issue to provide a user friendly and workable ad-hoc solution is to allow network nodes to interoperate seamlessly-without any user configuration. In the terms of autoconfiguration in ad hoc networks, service discovery (or resource discovery) is an important area. Service discovery provides functionality to automatically discover capabilities and advertise own capabilities to the network. Using service discovery, users can search for services by name, type or class and utilize those services without any further knowledge about the underlying network architecture.

Most of the typical ad-hoc scenarios can profit by service discovery technology: (a) In search, emergency and rescue operations, services can be patient databases, avalanche probe line equipment, sensors, gateways; (b) in vehicle networks, services can be news and weather forecast databases, transmission of road conditions, alarm subscriptions, map servers, location dependent travel guide; (c) in tactical networks, services can be gateways to higher lever echelon, combat id systems, sensors like cameras, infrared detectors, medical supervision of soldiers, position and location units; (d) in virtual classrooms, services can be printers, electronic whiteboard sharing, file sharing, Internet access, educational applications; (e) in the entertainment industry, services can be different part of ad hoc multi player games in airplains or trains, social interaction or so-called wireless dating. This is just and excerpt and the range of possible applications and services are enormous.

Service discovery scenarios is often exemplified by a user browsing a local area network for available printers. But as described above, service discovery is an ingenious solution for many other applications outside the office environment. As an example, the usefulness of service discovery in a military network can be illustrated with the following scenario: A soldier equipped with a battlefield PDA and a combat net radio establishes an ad-hoc network in cooperation with other soldiers and vehicles. Once the soldier is authenticated and authorized, he is able to browse the network for available services. Using the automatic service discovery protocol he is now able to discover two map servers placed in two different combat vehicles. The map application automatically utilizes those services to download useful electronic maps of the battlefield. The soldier also discover one of the local unmanned ground vehicles (UGV) with an IP-based camera and infrared detectors which the soldier can put to use. Further, the soldier carried PDA discovers medical sensors placed in the soldiers vest and armor and provides them as services, later utilized by the platoon commander and the medical corps. With the underlying ad-hoc routing protocol the described services and information is accessible using direct links or via intermediate nodes despite high mobility during the mission.

Even if the above example is an imaginary scenario, it is actually achievable using ad-hoc technology and service discovery protocols. The rest of this paper will discuss proposals on how to achieve service discovery in ad-hoc networks, and address some of the imperatives and challenges in this area.

3.2 Mechanisms

The different service discovery approaches can be categorized into three groups:

1. Service coordinator based architecture (centralized). Services are registered at a central entity, and clients search for services in this registry. This is usually the deployed architecture in fixed Ethernet networks.
2. Distributed query-based architecture (decentralized). Services are advertised to all clients, and service information is stored on each node that is offering the service.
3. Hybrid architecture. Service information is stored on each node, but a set of service coordinators is chosen to be the main binding between services and service requests.

Many proposals exists for utilizing the different approaches for service discovery. Some authors swear by a centralized architecture, while others find the decentralized approach attractive. According to [6], the decentralized architecture is the preferred approach for ad-hoc networks, while others [16] claims that the hybrid solutions are feasible. However, a lot of aspects determine the choice of solution. The size of the network, the number and type of services, service availability demands and the chosen ad-hoc routing protocol are all factors influencing the choice of service discovery protocol. As elsewhere in life, no size fits all.

3.3 Solutions

Several consortiums, companies and organizations has been doing research in the field of service discovery. Even if IETF has proposed a standard (SLP), the overall Internet community has not yet reached a consensus on one particular service discovery protocol. Although most of the proposed standards does not fit well in the ad-hoc environment, we will give a short introduction to the most popular solutions for service discovery over IP based networks. The most promising solutions are Service Location Protocol (SLP), DNS Service Discovery (DNS-SD), Simple Service Discovery Protocol (SSDP), Jini, and Bluetooth Service Discovery Protocol (SDP).

Service Location Protocol (SLP)

Service Location Protocol [8] is developed by IETF as a vendor independent standard. The SLP architecture is based on three components: (i) User agents - which is the software entity that performs the service discovery; (ii) Service agents - which advertises the location of services; (iii) Directory agents - which acts as a central repository and collects service information from service agents and responds to service requests from user agents.

The Service Location Protocol is not widely supported, mainly because dominant companies such as Apple and Microsoft are developing and supporting other service discovery protocols.

Simple Service Discovery Protocol (SSDP)

Simple Service Discovery Protocol (SSDP) is a part of Microsoft's UPnP [1]. UPnP is used in Windows XP and several brands of network equipment. Some manufacturers of network devices has included the UPnP protocols, which gives a zero configuration solution if used in a network with MS Windows computers.

SSDP utilizes Hypertext Transfer Protocol Unicast (HTTPU) and Hypertext Transfer Protocol Multicast (HTTPMU). The architecture is based on the following three components: (i) SSDP service - which represents the service agent; (ii) SSDP client - which is the user agent utilizing the services; (iii) SSDP proxy - which is the directory agent representing the binding between the SSDP service and SSPD client. The SSDP proxy is not a mandatory part of SSDP, meaning that service information can disseminate in the network without this central entity using multicast.

DNS Service Discovery (DNS-SD)

Apples *Bonjour* (formerly Rendevouz) is included in MAC OS X and is used by Apple software such as iPhoto and iTunes. It is also supported by the KDE and Gnome desktop environments. The system uses a combination of link-local address choosing [2], Multicast DNS (mDNS), and DNS-SD.

DNS Service Discovery is a way of using the existing DNS records to locate services. Since a ZeroConf (or Bonjour) implementation must likely will have a multicast DNS responder for the name-to-address translation, the amendment of service discovery can be implemented in quite a lightweight manner. DNS-SD is considered simpler than SSDP because it uses DNS rather than HTTP. The protocol can be used to obtain names, service type, port numbers and other attribute information.

Since Apple first launched Bonjour in 2002, every major maker of network printers has adopted Bonjour and uses DNS-SD to advertise the printer service to the local area network.

[1]The UPnP Forum is an industry initiative designed to enable simple and robust connectivity among stand-alone devices and PCs from many different vendors.

Jini

Jini[2] is an extension to Java, and allows Jini capable devices to form an ad hoc network and share resources and services in between. Jini allows clients to join a Jini lookup service (JLS). The lookup service correspond to the directory agent in the SLP protocol. Using the JLS, the clients can request information about services as well as publish own services. The lookup service can also store Java code necessary for the clients to access one particular service.

The fact that Jini is tied to the Java programming environment and Java Virtual Machine (JVM) makes it not suitable for every low powered embedded system.

Bluetooth Service Discovery Protocol (SDP)

The bluetooth communication stack contains the Service Discovery Protocol (SDP). The protocol addresses service discovery specially for bluetooth networks, and supports searching for services by *name*, searching by *attributes*, and browsing the network for *any* service. SDP is a scaled down solution, and does not address functionality on how to access the services. Thus, an additional protocol is usually built on top of SDP.

4 Service discovery in MANETs

We know the characteristics of mobile ad hoc networks, and remember that the idea behind MANET is based on a non-infrastructure approach. MANETs are mobile in nature, prone to network failures, and mobile nodes may leave or enter the network at any time. The beforementioned service discovery architectures are primarily designed for fixed networks, and due to the special characteristics of mobile networks, they might not be suitable for MANETs. Most of the proposed service discovery solutions is based on a service coordinator based architecture as a mode of operation. In such a centralized scheme, as outlined in section 3.2, services are registered in a central entity, and clients search for services in this central registry. A distributed or hybrid scheme is more suitable for MANETs as a fixed infrastructure cannot be assumed, and hence the link to the service coordinator may be unpredictable.

Besides from the three service discovery categories presented earlier (centralized, distributed, hybrid), service discovery mechanisms for MANETs can be further subdivided as follows: (i) General mechanism independent of the underlying routing protocol; (ii) mechanism integrated with a *reactive* routing protocol; (iii) mechanism integrated with a *proactive* routing protocol. Most of the MANET SD proposals belongs to the first category, and places service discovery at a layer above routing. However, since an ad-hoc routing protocol such as OLSR or AODV must be present in an ad-hoc network, it is convenient to integrate service discovery with the routing protocol.

In the following, a selection of different service discovery articles will be presented in short.

[2]http://sun.com/jini

4.1 Multimode routing

In the article “Analysis of decentralized resource and service discovery mechanisms in wireless multi-hop networks” [11], J.Hoebeke et.al presents a distributed service discovery architecture using AODV and WRP as reactive and proactive routing protocols respectively. The article discusses the tradeoffs between centralized and decentralized service discovery and evaluates the performance of both reactive (query based) and proactive (announcement based) service discovery solutions. The article concludes that a tight interaction between the routing protocol and the service discovery protocol is of great importance to ensure high performance, high service availability and to minimize delay. A performance evaluation study is conducted using GloMoSim network simulation in addition to analytical analysis.

The most interesting aspect in the article is the proposal to use multimode routing [10], and to integrate service discovery to the multimode solution. The multimode routing architecture dynamically switches between different routing schemes, i.e reactively and proactively, and changes its behavior according to changes in the network or service context.

Although the multimode or hybrid routing idea, introduces a great many challenges, the approximation of a integrated solution of the service discovery and routing shows that a deep understanding of the routing protocols is mandatory. The two next subsections outlines solutions which integrates service discovery in reactively and proactively routed networks respectively.

4.2 Reactively routed MANETs

Engelstad et.al [7] evaluates the performance of the hybrid and the distributed scheme in a reactively routed MANET using AODV. The simulations is conducted using the GloMoSim network simulator. All the simulations are done with a network of 50 nodes. The article shows how both a distributed and a hybrid service discovery architecture can be integrated with the AODV routing protocol. Service discovery requests (SREQ) are piggybacked on routing request packets (RREQ), and service discovery replies (SREP) are piggybacked on routing reply (RREP) packets. In the hybrid architecture, the service coordinator announcements are piggybacked on RREQ packets and service registrations are piggybacked on RREQ packets, figure 1.

The simulations shows that the distributed query based service discovery scheme outperforms the hybrid scheme in reactively routed MANETs. The hybrid architecture puts an extra load to the network, as it triggers additional route requests- and replies compared to the distributed architecture. Obviously, this is not the case in proactively routed networks, where route requests are not on demand per nature. Moreover it is pointed out that hybrid architectures introduces complicated mechanisms for electing service coordinators, and puts extra load to the infrastructure. One of the simulations shows that increased network dynamics (in the case of mobility, entering or leaving nodes) increases the amount of false positive service replies from the service coordinators.

The article suggests the investigation of caching of service descriptors as a basis for future work. Local caching seems to enhance the distributed architecture with the benefits

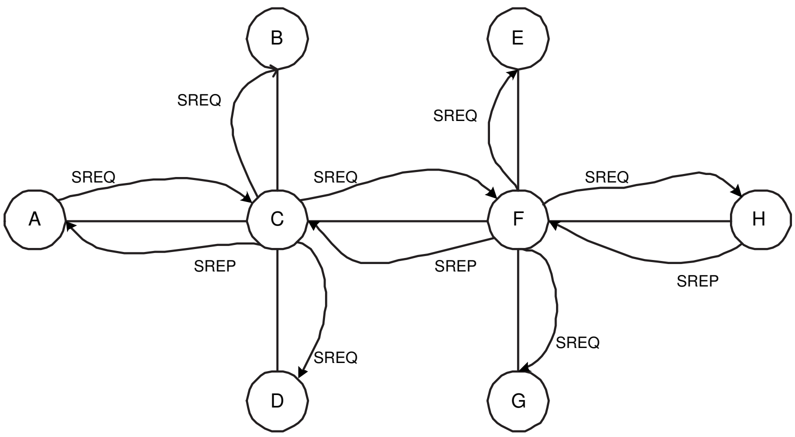


Figure 1: Distributed query-based service discovery. Service discovery requests (SREQ) are piggybacked on routing request packets (RREQ), and service discovery replies (SREP) are piggybacked on routing reply (RREP) packets.

of the hybrid architecture.

4.3 Proactively routed MANETs

Jodra et.al [12] presents a novel solution where service discovery is integrated with OLSR. The different OLSR messages as outlined in section 2.2.2 shares a common message header. Utilizing this header, a new message can be described. The new message is called Service Discovery Message (SDM). The SDM packet can contain either a service advertisement or a query. An example of service advertisement using OLSR piggybacking is outlined in figure 2. The proposal also introduces a service cache for each node in the network. The cache will store all services which is available for the node, both local and foreign. Whenever a node wants to use a service not stored in its local cache, it will send a query asking for that particular service using the SDM query message. The SDM messages will be forwarded by the MPRs in the network. Upon receiving such a SDM query message, a node will check whether its local services corresponds to the service asked for in the query SDM. If this is the case, it will send an advertisement message announcing the requested service. The answer will be MPR flooded.

The paper further describes some simulation results using Network Simulator 2 (ns-2)[3] including the proposed SDM extension using the UM-OLSR plugin from University of Murcia. Both a large and a small scenario is simulated. Unsurprisingly, as the complete SDM messages is only 8 bytes, and thanks to piggybacking of the SDM to the OLSR packets, the simulations shows that SDM adds a very small overhead to the network. However, a comparison with other service discovery solutions is not included in the paper.

Overall, this is a good proposal, however, some work is still remaining. The SDM message is extremely slim, and a service name of only 8 bits is not a good foundation

[3] http://www.isi.edu/nsnam/ns/

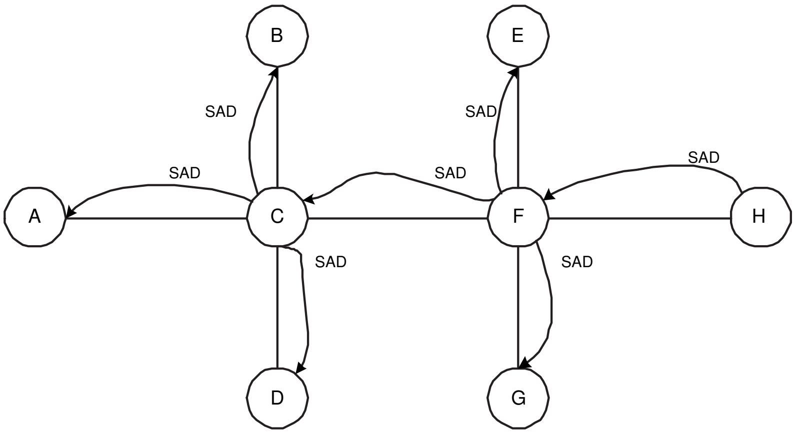


Figure 2: Distributed proactive service discovery. The MPRs in the OLSR network are used to disseminate service advertisements (SAD).

for a flexible solution. In addition, a mechanism on how to convert human readable service names and attributes to the scaled down SDM message is not a part of the paper. A Bloom filter (see figure 3) is not, but should be considered. The time a valid service advertisement should stay in the local cache before it is removed should be further evaluated. The information in the cache is only valid for a limited time, and a long timeout value will possibly lead to erroneous entries in the cache, and on the other hand, a short timeout will lead to increased overhead due to repetitive queries.

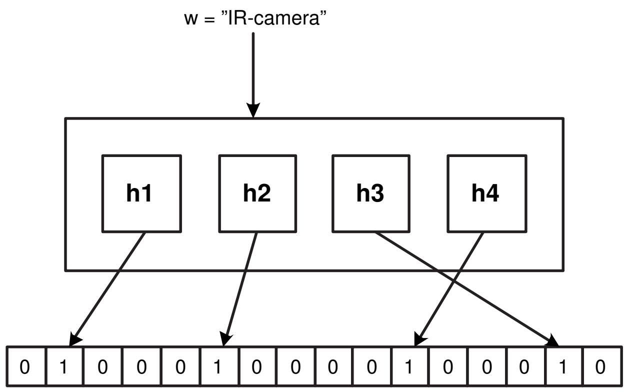
A similar proposal using OLSR piggybacking is described in Li and Lamount's article "A lightweight service discovery mechanism for mobile ad hoc pervasive environment using cross-layer design"[13]. A message similar to the previously mentioned SDM, Service Location Extention (SLE), is here introduced. However, the technical details about the format are left out of the paper.

4.4 Directory based service discovery

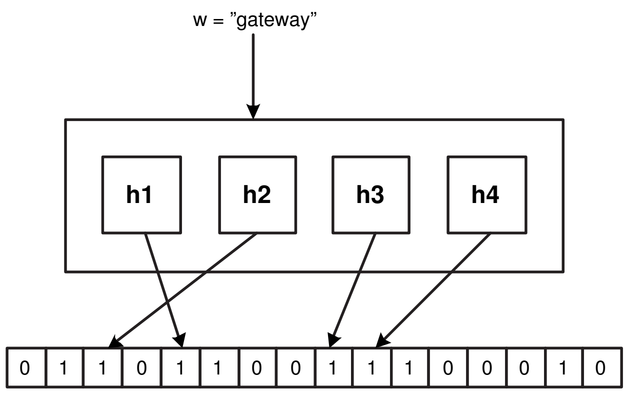
In the paper "Scalable service discovery for MANET" [16], Sailhan and Issarny proposes a directory based service discovery architecture. The work is based on a centralized discovery architecture, as this, according to the authors, induces less traffic. The directories consists of a collection of MANET nodes and a service coordinator. Directories are distributed and deployed dynamically, and forms a virtual backbone of nodes. Communication between directories is done using bordercasting instead of flooding. The proposed bordercast technique works exactly like MPR selection and MPR flooding as used in the OLSR routing protocol. Even if the proposal outlines an architecture independent of the routing protocol, simulations described in the paper utilizes OLSR as routing protocol and replaces the proposed bordercast realization protocol by taking advantage of the OLSR MPRs. The use of directories seems to be more applicable in very large networks, though, the number of nodes consisting in the simulations vary from 10 to a maximum of 100.

The proposed solution spins down to two distinct optimizations: (i) In contrast to [12], which defines a short predefined bitstream for service description, Sailhan et.al proposes the use of Bloom filters as an efficient way to describe the services, based on hashing the service descriptions; (ii) the overall traffic is minimized thanks to the use of bordercasting.

To sum up, this article emphasizes the importance of a tight integration of the routing protocol and the service discovery architecture, and shows that the use of MPR flooding is a clever solution to disseminate service information in an OLSR network. The proposal to use Bloom filters to describe services are also proposed by others[14] [9]. A Bloom filter example is shown in figure 3.



(a) IR-camera is added to the Bloom filter



(b) gateway is added to the Bloom filter

Figure 3: Two services added to a Bloom filter v of m bits initially set to 0. Service descriptors are run through k independent hash functions $h_1, h_2, \ldots h_k$. For each service description $w, v[h_i(w)] = 1$ where $i = 1, 2, \ldots k$

5 Summary

Some of the studies presented demonstrates that the use of service coordinators, i.e a hybrid architecture, will increase service availability. However their benefits are unclear, and the increase in service availability may be negligible compared to the extra message overhead introduced. However, local caching of service descriptions seems to combine the benefits of a hybrid architecture with the benefits of the distributed architecture. The use of caching should therefore be further studied.

Even if the majority of the service discovery solutions and proposals address service discovery at the application layer, a cross-layer integration of the service discovery architecture with the routing protocol seems to bring considerable optimizations. Although this solution will violate the modular OSI layered approach and possibly hinder easy interchangeability of routing protocols, the benefits are indisputable both in proactively and reactively routed MANETs.

Several articles shows that the use of Bloom filters, based on hashing the service descriptions, is an efficient way to describe the services. In stead of disseminate full service information, the use of Bloom filters will save a lot of network resources, which is sparse in ad hoc networks. As Bloom filters is built on a one-way hash function, it is always a chance for false positive service replies. False positives will occur if two or more service descriptors hashes to the same value in the Bloom filter. On way to reduce the chance of false positives, is to enlarge the Bloom filter. Hence, to discover the optimal filter width is crucial to fully take advantage of the Bloom filtered service discovery mechanism.

To sum up, the three most important lessons from this tiny study are: (i) a distributed (decentralized) service discovery architecture is probably the most convenient solution for mobile ad hoc networks. The use of local caching seems to increase the performance of the distributed scheme; (ii) a tight integration of the service discovery protocols and the MANET routing protocol is crucial for optimal overall performance; (iii) the use of Bloom filters is an efficient and compact way to describe services.

5.1 The authors suggestion for a future solution

The authors suggestion is to combine the three bullet points described above width the latest advances in the MANET group. A fully distributed scheme with local caching using a proactively routing protocol is a reasonable starting point for a high performance service discovery protocol.

A promising solution is to base the service discovery protocol on the latest MANET Packet Building Block (PacketBB) [3], supported in OLSRv2 and DYMO. The PacketBB format makes it possible to use OLSRv2 or DYMO to disseminate service discovery messages. However, the most important feature using PacketBB based service discovery is the fact that it makes it possible for nodes not supporting or participating in the service discovery network (i.e not running any service discovery code) to forward service requests and service replies on behalf of participating nodes. In addition, the system will still have the optimization benefits of an integrated (cross-layer) routing- and service discovery solution.

An other similar approach is to use the OLSRd[4] implementation, and to add a service discovery format using the included plugin-interface. The packet format as described in the previously mentioned article[12] could be a reasonable starting point.

Further, the service information should be based on both a optimized description such

[4] www.olsr.org

as using Bloom filters and a more descriptive language for infrequent updates. The latter solution makes it possible to add more detailed service descriptors necessary for some applications and services.

Even if the Internet community has not yet reached a consensus on one particular service discovery protocol for Internet, the opportunity to create a compatibility layer to one of the employed standards should be considered. The DNS-SD standard used in MAC OS X, KDE and GNOME desktops is, although not directly applicable to MANET, a promising and lightweight solution for Ethernet. A compatibility layer between a future MANET service discovery architecture and DNS-SD will allow existing applications to work seamlessly over a MANET.

References

[1] I. Chakeres and C. Perkins. Dynamic manet on-demand (dymo) routing, May 2007. INTERNET-DRAFT draft-ietf-manet-dymo-09.

[2] S. Cheshire, B. Aboba, and E. Guttman. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927 (Proposed Standard), May 2005.

[3] T. Clausen, C. Dearlove, J. Dean, and C. Adjih. Generalized manet packet/message format, January 2007. INTERNET-DRAFT draft-ietf-manet-packetbb-04.txt.

[4] T. Clausen, C. Dearlove, and P. Jacquet. The optimized link state routing protocol version 2, February 2007. INTERNET-DRAFT draft-ietf-manet-olsrv2-03.

[5] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.

[6] P. E. Engelstad and Y. Zheng. Evaluation of service discovery architectures for mobile ad hoc networks. In *WONS '05: Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05)*, pages 2–15, Washington, DC, USA, 2005. IEEE Computer Society.

[7] P. E. Engelstad, Y. Zheng, R. Koodli, and C. E. Perkins. Service discovery architectures for on-demand ad hoc networks. *International Journal of Ad Hoc and Sensor Wireless Networks, Old City Publishing (OCP Science)*, 2(1):27–58, March 2006.

[8] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service Location Protocol, Version 2. RFC 2608 (Proposed Standard), June 1999. Updated by RFC 3224.

[9] Robbert Haarman. Ahoy: A proximity-based discovery protocol. Master's thesis, January 2007.

[10] J. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester. Adaptive multi-mode routing in mobile ad hoc networks. *Personal Wireless Communications*, pages 107–117, 2004.

[11] J. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester. Analysis of decentralized resource and service discovery mechanisms in wireless multi-hop networks. *Computer Communications*, 29(13-14):2710–2720, August 2006.

[12] J. L. Jodra, M. Vara, J. M. Cabero, and J. Bagazgoitia. Service discovery mechanism over olsr for mobile ad-hoc networks. *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, 2:6 pp.+, 2006.

[13] Li Li and L. Lamont. A lightweight service discovery mechanism for mobile ad hoc pervasive environment using cross-layer design. *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 55–59, 2005.

[14] F. Liu and G. J. Heijenk. Context discovery using attenuated bloom filters in ad-hoc networks. 3970:13–25, May 2006.

[15] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[16] F. Sailhan and V. Issarny. Scalable service discovery for manet. *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 235–244, 2005.