Below Cross-Layer: An Alternative Approach to Service Discovery for MANETs

Warren Kenny and Stefan Weber

Trinity College, College Green, Dublin, Ireland
`kennyw@scss.tcd.ie,sweber@scss.tcd.ie`

Abstract. Service discovery protocols for mobile ad hoc networks attempt to overcome the inability to locate resources presented by networks in which prior knowledge of node identity and capability is not available. Existing approaches continue to rely on underlying address-based routing protocols in order to communicate with discovered services. These two-tier approaches generate routing overheads which negatively impact on performance and network scalability.
As high-powered mobile computing devices with wireless connectivity become increasingly ubiquitous, the need for routing protocols which can operate at increased network densities becomes more acute. Cross-layer approaches to service discovery in MANETs have attempted to optimize the discovery process through direct integration with underlying routing protocols, however additional steps are necessary to improve service discovery performance, network scalability and application throughput. This paper describes the Service Discovery and Routing Protocol (SDRP), a novel service-oriented routing protocol for MANETs. This protocol eschews the use of network-wide unique addresses or underlying address-based routing protocols and focuses instead on routing only to and from nodes which provide services. A comparison with existing approaches demonstrates that this approach improves discovery success rates and application throughput at higher node densities.

Key words: MANET, Routing, Service Discovery

1 Introduction

MANETs are characterised by variable topologies, unreliable connections between nodes and limited resources [6]. It is assumed that nodes have limited battery life and communication capacity, thus it is important to minimise unnecessary overheads in order to maximise performance.

Existing research in the area of routing for mobile ad hoc networks (MANETs) has focused on a particular set of deployment scenarios; disaster zones, battlefields and wireless sensor deployments. As high-powered mobile devices with wireless capability become increasingly ubiquitous, new scenarios involving drastically greater network densities and sizes are set to emerge; particularly those in urban environments. Such scenarios involve network scales far beyond those traditionally studied [1].

Current approaches to routing are designed to replicate the functionality of infrastructure networks [14], particularly facilitating arbitrary connections between nodes based on addresses. Infrastructure networks are capable of supporting reliable communications between arbitrary nodes with routing based on optimal paths over dedicated routers along high-bandwidth links. However, providing this functionality in MANETs has an adverse effect on communication in terms of the overheads that are generated and the interference that this causes [4, 9] and may not be well suited to typical MANET deployment scenarios such as those mentioned above.

Facilitation of arbitrary connections between nodes also makes little sense in the above MANET deployment scenarios, where the capabilities and identities of nodes are unlikely to be known ahead of network deployment. Service-oriented routing protocols [20] attempt to solve this problem by allowing services provided by nodes in the network to be discovered or advertised to peers. However, these approaches assume that arbitrary connections between nodes are already facilitated in the network by an underlying routing protocol and thus attempt to provide nodes with connections to services by mapping service requests to network addresses, creating a two-tier routing approach which generates unnecessary overheads and suffers from the same problems as address-based approaches.

In this paper we present a highly-scalable and flexible service-oriented routing protocol for MANETs called Service Discovery Routing Protocol (SDRP). SDRP does not use an underlying address-based routing protocol, but was instead designed to provide routing only to services rather than to arbitrary node addresses. This approach, combined with a novel use of Bloom Filters [2] for the purpose of optimizing service advertisement, results in a routing protocol which exhibits favourable characteristics in a variety of scenarios; particularly in large-scale, high-density networks.

The rest of the paper is organized as follows: Section 2 will discuss protocols which implement service-discovery architectures mentioned above. Section 3 discusses the design and implementation of SDRP. Section 4 presents an analysis, through simulation, of SDRP and a number of service-oriented protocols. Sections 5 will discuss the results of this analysis and the conclusions that can be drawn from the performance of the protocol.

2 Related Work

The following section discusses a number of service-oriented routing protocols for MANETs. These protocols may be classified as either service coordinator, distributed query-based or hybrid architectures [17].

Service-coordinator protocols designate a subset certain nodes as Service-Coordinators (SCs) or directories which are responsible for tracking the services provided by Service Agent (SA) nodes [15]. User Agent (UA) nodes connect to SCs in order to request services. SCs then respond with a list of SAs which provide that service. Directories advertise their presence either through direct advertisement flooding through a cross-layer approach based on the piggy-backing advertisements onto routing protocol packets.

Such protocols have been shown to perform well with large numbers of service agents and user agents, as advertisement flooding is restricted to service coordinators which can be distributed in optimal numbers during deployment. Directory placement is, however, a key concern as service location delay and success rates depend on distance between service requesters and service directories [7].

Hybrid approaches solve this problem by falling back on direct query flooding or service advertisement when service coordinators are unavailable [12], thus improving service availability in networks with high node failure rates or frequent partitioning.

Distributed query protocol operate in a purely peer-to-peer fashion; with clients querying the network for servers or with servers advertising their services to the network depending on the specific protocol mechanism [21, 3]. These approaches may generate high overheads when compared to service coordinator approaches, as queries and advertisements are directly flooded throughout the network by user agents or service agents.

These categories can be further sub-divided into cross-layer and application-layer architectures [19]. Cross-layer approaches attempt to optimize communications by integrating with an underlying routing protocol [18], piggy-backing service discovery messages or advertisements onto routing messages. Application layer protocols operate independently of the installed routing protocol, advertising or flooding service requests and advertisements directly in order to discover the addresses of servers. This disconnect results in some inefficiency due to redundant broadcasts and transmissions, however application-layer approaches can operate on top of any routing protocol, unlike cross-layer approaches.

Mercury [5] is a cross-layer protocol integrated with OLSR [11]. OLSR attempts to optimize dissemination of link state information using its multi-point relay (MPR) algorithm, which reduces flooding by selectively rebroadcasting topology control message based on link-state data received from neighbours in periodically broadcast HELLO messages. OLSR attempts to determine the subset of neighbour nodes required to reach all two-hop neighbours. Those neighbours are then selected as multi-point relays and rebroadcast topology control messages containing the source's link-state information. This approach improves on naive flooding mechanisms be eliminating redundant rebroadcasts. Mercury piggybacks service advertisement messages onto OLSR's topology control messages and takes advantage of the OLSR's extensible messaging format by appending a *Service Filter*; a bloom filter [2] containing service descriptor strings for local and neighbour node services.

Service descriptors are encoded using the MD5 digest algorithm [16], producing a 128-bit hash value. This hash value is split into k groups of r bits, resulting in offsets into the bloom filter to be set in order to encode the service descriptor. When a service advertisement is received, the receiving node adds the source node's address and associated service filter to its service routing table. When an application requests a service, this routing table is checked by first encoding the requested service's descriptor and checking it against stored filters. If an associated address can be found, the message is then forwarded using OLSR.

Mercury is designed to reduce the overheads associated with service discovery. Mercury's use of a cross-layer design reduces messaging overheads as service advertisements are attached to routing protocol packets instead of being sent separately. The use of bloom filters as service descriptors acts as an optimization in terms of header size when compared with direct linear serialization of service information, particularly when service information is in the form of variable-length strings or complex data structures. As OLSR is a proactive routing protocol, its generated overheads can be expected to increase with increasing network size as additional nodes send HELLO and topology control messages.

The Lightweight Service Discovery protocol (LSD) is a cross-layer service coordinator discovery mechanism designed to adapt to changes in the underlying network and reduce messaging overheads through direct interaction with an underlying proactive routing protocol [12]. Nodes in an LSD network are designated as either clients, service nodes or directories. Service nodes register their network addresses and a description of their services with available directories in order to allow client nodes to locate them. These registrations are refreshed periodically in order to avoid false positives during service requests. Directory nodes periodically advertise their presence in the network by piggybacking advertisement messages onto routing protocol topology control messages. Client nodes cache addresses for directory nodes and may query them in order to locate service nodes as required.

LSD integrates with OLSR, allowing it to attach directory advertisements to OLSR topology control message broadcasts. This approach reduces overheads by combining route and service information into a single packet rather than multiple separate packets. LSD is designed to adapt to changing network conditions by falling back on a direct query broadcast approach when no directory node is available. Client nodes piggyback a query for a service onto topology control messages as described above, allowing clients to locate services without the assistance of an intermediate directory node. However, as with Mercury, the use of OLSR as an underlying routing protocol can be expected to result in higher overheads as network density increases. In addition, optimal distribution of directory nodes is required in order to avoid the use of direct query flooding; a difficult task in large-scale or highly mobile networks [7].

SMF [21] is a cross-layer distributed-query protocol based on AODV [13] which combines aspects of advertisement and query protocols; advertisements are piggybacked onto AODV HELLO messages and distributed to 1-hop neighbours while AODV's route request messaging mechanism is extended to include a service request function. SMF uses received HELLO and route response packets to calculate a metric called the service magnetic field. This metric is used for route selection and is based on the number of hops to the server as well as the density of servers providing the same service in a given area. SMF's reactive query-based approach performs well in networks where the number of clients is expected to be low, as AODV's flooding approach to route discovery can overwhelm available bandwidth when there are many service or route requests [8].

The Pervasive Discovery Protocol (PDP) is an example of an application-layer protocol. PDP does not integrate with the underlying routing protocol, instead assuming that a multicast-capable routing protocol is available. PDP uses a hybrid pull-push service advertisement feature that delays advertisements for services until they are first requested; thus ensuring that only service which are in demand use network resources. PDP's application-layer design allows it to be deployed in any MANET with a multicast-capable routing protocol installed, however this lack of integration may result in additional overheads and poor mobility adaptation when compared to the cross-layer approaches described above.

2.1 Discussion

The above approaches attempt to map services to network addresses which are then be used by an underlying routing protocol to direct packets to a discovered server. By tightly coupling services to network addresses, existing protocols reduce potential redundancy and add an unnecessary additional step in the routing process.

Once the described protocols have mapped service identifiers to network addresses, their role in the process is finished. At this point, the underlying routing protocol is responsible for routing traffic to and from the selected server. In the case where the server is no longer available, due to network partitioning or battery depletion for example, the path between the client and server is lost and the discovery process must be restarted at the source. In addition, the overheads generated by the address-based routing step can negatively affect application throughput, particularly in high-density networks [1].

These factors combine to reduce the usability of service-oriented networks based upon such protocols. The next section will detail how SDRP addresses these issues.

3 Service Discovery Routing Protocol

Service Discovery Routing Protocol (SDRP) is designed to provide robust, low-overhead routing to and from servers in MANETs. It accomplishes this through a combination of header size reduction through extensive use of bloom filters, the use of a mechanism similar to OLSR's MPR [11] for proactive service advertisement, the removal of support for arbitrary node-to-node routing and by decoupling network addresses from service descriptors.

3.1 Overhead Reduction and Robustness

SDRP reduces overhead generation and improves scalability when compared to alternative approaches which rely on an underlying routing protocol. In a network based upon an address-based routing protocol, nodes generate control overheads regardless of their role or functionality. In the case of OLSR, all nodes generate periodic HELLO messages, while topology control messages are flooded by MPR nodes. Reactive protocols such as AODV flood route requests [13]. In an SDRP-based network, only server nodes act as sources of advertisements. This approach retains the advantage of a proactive protocol, such low end-to-end delay and prior knowledge of existing services, while reducing overheads by restricting message flooding to a subset of network nodes based on role.

SDRP adds redundancy to the routing process by decoupling network addresses from service identifiers. At each hop in a multi-hop client-server connection, packet headers are inspected in order to ascertain the target service. Each node on the route sends the packet to the next hop on the shortest route to the target service, as recorded in its service routing table. This approach allows service requests to follow shortest paths to an available server and be re-routed by intermediate nodes when that server is no longer available.

3.2 Header Size Reduction

Bloom filters are used extensively by SDRP in order to reduce network overheads. However unlike Mercury, which uses them only for optimizing the transmission of service descriptors, SDRP also uses them to convey link state information in order to optimize service advertisement distribution.

Bloom Filters [2] are space-efficient probabilistic data structures used to represent sets, against which membership queries may be performed. A bloom filter is a bit array of m bits initially set to 0. Insertion is performed by executing k hash functions on elements in order to obtain array offsets. These offsets are then set to 1.

A membership query is performed by using the same process to generate array offsets and then checking whether each offset is set to 1. If any of the produced offsets are set to 0, the element is definitely not in the filter. If the offsets are set to 1, it is assumed that the element is present in the filter, although there is a probability that this is incorrect.

The probability of incorrect membership queries increases as more elements are inserted into the filter and more filter bits are set. Eventually, a filter may have all of its bits set, in which case all membership queries will succeed and the filter becomes functionally useless. The probability of a false positive query occurring is a function of the array size m, number of inserted elements n and number of hashes used k. The probability that a specific bit is still 0 after n elements have been inserted into the filter may be expressed as

$$\left(1 - \frac{1}{m}\right)^{kn} \approx e^{-kn/m}$$

Thus there are tradeoffs between the computational effort in terms of calculation of k hashes per element, bit array size and the probability of a false positive occurring. Bloom filters have a strong space advantage over other data structures for representing sets. A bloom filter with an optimal hash count and a 1% false-positive rate requires approximately 9.6 bits per element, regardless of the size of the elements. Compared to direct serialization of network addresses or service descriptors, bloom filters represent a space-efficient method for containing set data; OLSR's linear serialization of network addresses, for example, requires 32 bits per address, as well as the space required to convey link-state. Thus the use of bloom filters for conveying information in protocol messages provides a significant advantage in terms of control overheads reduction.

3.3 SDRP Advertisement Mechanism

Servers in an SDRP network advertise their services using an algorithm called Reduced MPR (RMPR). RMPR is is designed to further reduce messaging overheads when compared to the MPR approach used by OLSR by avoiding the broadcast of HELLO packets and using dynamically-sized bloom filters for neighbourhood description instead of direct neighbour link-state serialization.

Servers periodically broadcast service advertisements containing a bloom filter containing the identifiers of the nodes in their local area and another filter, the MPR filter, containing the identifiers of those nodes that should rebroadcast the packet. Nodes receiving this advertisement will rebroadcast if their address is contained in the MPR filter or if their address is not contained in the neighbourhood filter. Thus the role of HELLO messages in describing the link state of all neighbouring nodes is replaced using service advertisement messages.

The use of dynamically-sized bloom filters for neighbourhood description means that bitwise comparison of filters for the purpose of MPR selection cannot be used. Instead, a probabilistic approach is used to calculate MPR nodes using the following steps:

1. Enumerate over all stored neighbours
2. Generate a bloom filter with the same parameters as the current node's neighbour filter
3. Insert the addresses of all neighbours except the current neighbour into the generated filter
4. Perform a difference operation on the filters and insert the result into an ordered container

The above algorithm produces a set of neighbours ordered by how different their neighbourhood is from the source node's. This difference may be interpreted as a measure of how distant a neighbour is from the source and how many nodes are in the neighbour's local area which are not also in the source's local area.

Neighbour nodes with a distance metric higher than a configured value are then added to the MPR set. The set is further filtered by determining which of the chosen nodes are likely to be near to eachother through comparison of their neighbour filters. Nodes which are determined to be in close proximity to eachother based on comparison of their neighbour filters are then removed from the MPR set.

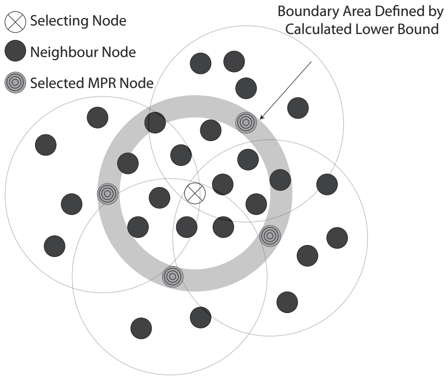

Fig. 1. Nodes are selected for the rebroadcasting of service advertisement based on their probable distance from the source determined by comparison of neighbour filters.

These steps remove nodes on the boundary of the local node's coverage which are within close proximity of eachother as illustrated in figure 1. Unlike OLSR's MPR mechanism, RMPR cannot reliably choose MPR nodes with unique connections to second-hop neighbours or an optimal set of MPR nodes to achieve full coverage due to the probabilistic nature of bloom filters. However, the control overhead reduction achieved through the use of variable-length bloom filters, removal of HELLO messages and selective rebroadcasting allow for greater scalability and throughput when compared to existing approaches. Calibration of the distance metric limit can increase MPR selection reliability at the expense of control overhead increase due to more frequent rebroadcasting.

Reduced MPR mode is designed to operate well in very high-density networks with low mobility, such as networks composed of pedestrian mobile devices in urban areas. The reduction in control overheads achieved by the removal of HELLO broadcasts and the size reduction in advertisement due to the use of dynamically-size bloom filters results in lower medium contention and improved application throughput.

4 Evaluation

SDRP is implemented as a shared library, written in C++, to allow for possible deployment on node hardware in the future. For the purpose of evaluation, it was linked with the NS-2 simulator [10]. Mobility was simulated using the random waypoint mobility model generator *setdest* which is distributed with NS-2. Comparison was performed against a number of alternative service discovery protocols. Mercury, PDP and SMF were included in the evaluation.

As an LSD NS-2 agent implementation was unavailable, it was reimplemented as an application-layer protocol operating on top of both AODV and OLSR. This implementation was named Service Coordination and Discovery Protocol

(SCDP). SCDP operates similarly to LSD. Directory nodes store the addresses of registered servers in the network and broadcast their presence periodically. User agents which cannot locate a directory node adapt by flooding requests for services as needed. However, SCDP does not integrate with the underlying routing protocol using a cross-layer design as LSD does. This design decision was made in order to expand the evaluation to include more protocols based on reactive routing mechanisms. The version of SCDP integrated AODV is called SCDPRA (Reactive) while the version integrated with OLSR is named SCDPPA (Proactive).

4.1 Experiment Setup

Client applications were simulated using a request-response agent designed specifically this evaluation. Limitations in NS-2's scripted agent connection mechanism mean that spontaneous connections between application agents at runtime are not well supported. This poses a problem in a network based on dynamic service discovery, thus the use of NS-2's built-in FTP and CBR agents was not possible. This may also be the reason for the lack of application throughput analysis in previous evaluations of the above protocols. The client application attempts to discover a service at set intervals and, if available, attempts to send packets to the located server. When the server application receives a request packet, a response packet is returned to the client. The client then immediately sends another request until the server can no longer be reached and the periodic request process begins again. The number of discovery attempts, requests sent, requests received, responses sent and responses received are recorded. The application's total data throughput is also measured.

Nodes are configured with a single wireless interface and omnidirectional antenna. Maximum wireless reception range was set to 160m, matching wireless interface parameters detailed in the technical specifications of an Orinoco 802.11b PC Card.

The evaluation aims to measure the overheads generated by the above protocols and measure the throughput the may be achieved by client nodes utilizing services in the network. Overheads are defined as the number of bytes of messaging traffic sent by the discovery protocol and, if applicable, its underlying routing protocol, per second. Throughput is defined as the number of bytes of application data received by servers in the network per second. Both overheads and throughput are measured as a combined total across all nodes and servers.

Service availability and service request message success rates are also measured. Service availability may be defined as the fraction of service discovery attempts which succeed. Service discovery involves the client application querying the discovery protocol for a route to a particular service. Depending on the protocol used, this may produce a flooded query or a check against a stored table. Service request success rate is defined as the number of application packets sent to the discovered server which are successfully received. Static simulation parameters are detailed in table 1.

Table 1. Static Simulation Parameters

Simulation Time	300s
Topography	Flat 1000m x 1000m
Mobility Model	Random Waypoint
Node Pause Time	5s
HELLO Message Interval	1s
Advertisement Cache Period	30s
TC/SA Interval	3s
Client Discovery Interval	1s
Service Count	3
Server Count	3
Client Count	5
Bloom Filter False Positive Probability	0.01

4.2 Experiment Results

Effect of Network Density The node count was varied between 10 and 200 nodes in order to measure protocol performance with increasing network density. Nodes are configured to move at 5m/s using the random waypoint model. A total of 3,200 simulation runs were used, with 20 runs performed per protocol-node count combination in order to reduced the effects of random variation. The number of client and server nodes remain static in this evaluation so that the effects of node and client/server density variations are not conflated in the results.

A comparison of application throughput with node count can be seen in figure 3. It can be observed that SDRP performs well in this scenario, with protocols based on application-layer or a proactive design experiencing a large drop in throughput with increasing node count. This may be attributed to the large increase in overheads experienced by such protocols with increasing network density, as illustrated in figure 2.

It is observed that the overheads generated by protocols based on OLSR are similar and increase greatly with increasing network density. This may be attributed to the increased number of HELLO and topology control messages sent as new nodes are added to the network. At a node count of 200, the overheads generated by these protocols dominate the medium and leave no bandwidth for application traffic. NS-2 is also designed to prioritize routing traffic over agent traffic, thus contributing to this effect. SDRP's overheads primarily increase as more servers are added to the network, thus allowing it to scale well when compared to address-based protocols. Reactive protocols such as SMF also perform well here as they only generate traffic in response to service requests. SDRP is observed to produce significantly lower overheads than all other protocols due to the lack of HELLO broadcasts and the reduced size of service advertisements due to the use of dynamic bloom filters.

Service availability is illustrated in figure 5. Here we see the protocols which use a cross-layer design such as Mercury and SMF perform best due to NS-2's

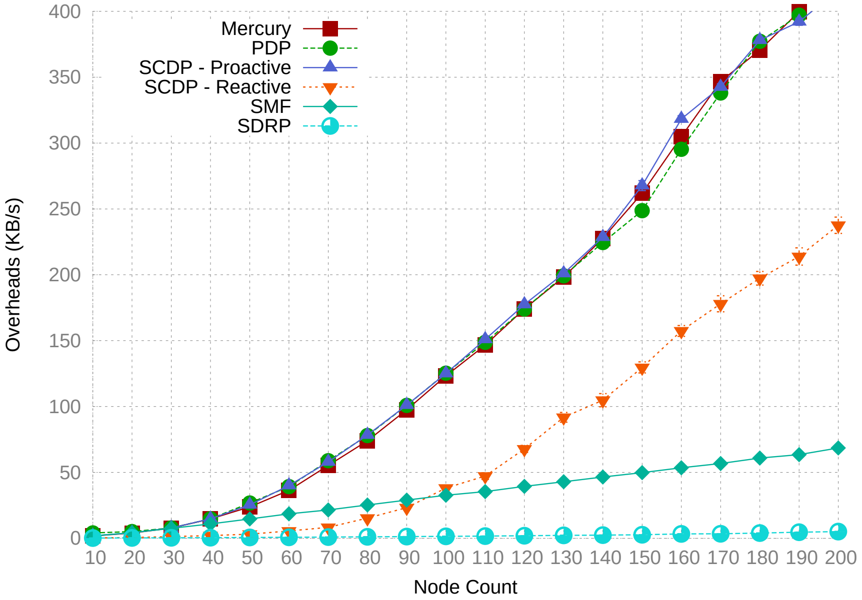

Fig. 2. Control Overheads vs. Node Count. As the network density increases, approaches based on an underlying routing protocol experience a significant increase in control overheads, while SDRP's remains low.

prioritization of routing traffic while application-layer protocols such as PDP and SCDP experience a drop in availability as medium contention increases. SDRP also performs well due to its lower messaging overheads.

Figure 4 shows the request success rate compared with node count. Here it can be seen that proactive and application-layer protocols experience sharp drops in success rates which correlate with the high overheads produced at greater network densities. As medium contention increases, the probability that an application packet will experience collisions or delays during transit increases. Due to the lower overheads produced by the reactive protocols and SDRP, success rates remain high. It is observed that protocols which achieved high availability rates, such as Mercury, suffer from low success rates; this is due to NS-2's prioritization of routing protocol traffic above data traffic. As service advertisements are carried in routing protocol messages, those messages are more likely to traverse the network than the data packets sent by client-server communication.

5 Conclusions

In this paper we presented a new service discovery protocol for MANETs called SDRP. SDRP was designed to maximise network scalability by reducing overheads through the use of compressed data structures and by avoiding reliance on

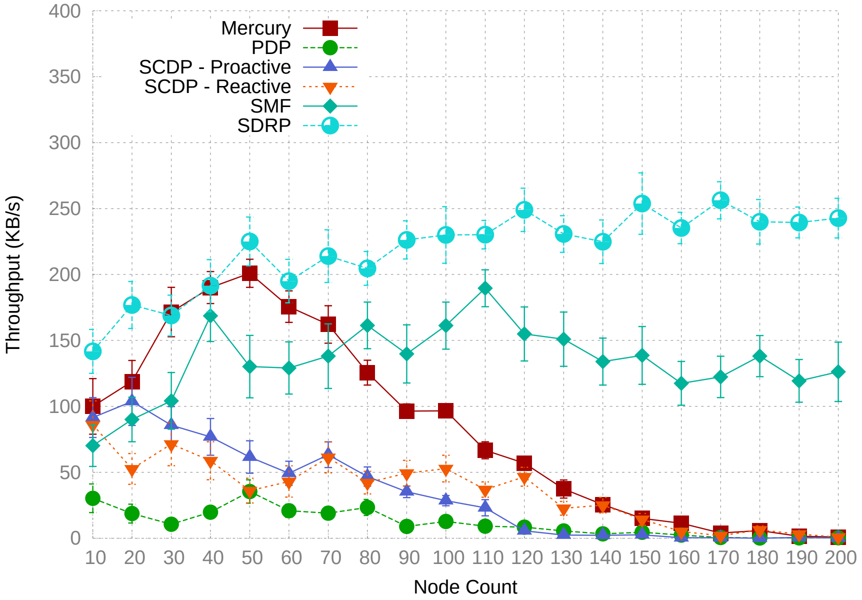

Fig. 3. Application Throughput vs. Node Count. The control overhead increase experienced by other approaches tends to have a corresponding negative impact on application throughput due to medium contention and interference. Here it can be seen that SDRP's low control overheads result in higher application throughput.

an underlying routing protocol. SDRP is designed to provide routing only to and from servers in MANETs, rather than between arbitrary nodes, and incorporates additional optimizations which enhance network robustness and communication reliability.

SDRP was evaluated against existing service discovery protocols using the NS-2 simulator. The evaluation demonstrates that SDRP significantly reduces overheads compared to existing approaches without affecting the success rates of service requests, thus improving application throughput; particularly at high network densities.

References

1. D. Arora, E. Millman, and S. Neville. Assessing the expected performance of the olsr routing protocol for denser urban core ad hoc network deployments. In *IEEE 26th International Conference onAdvanced Information Networking and Applications, AINA 2012*, pages 406–414, March. 2012.
2. B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

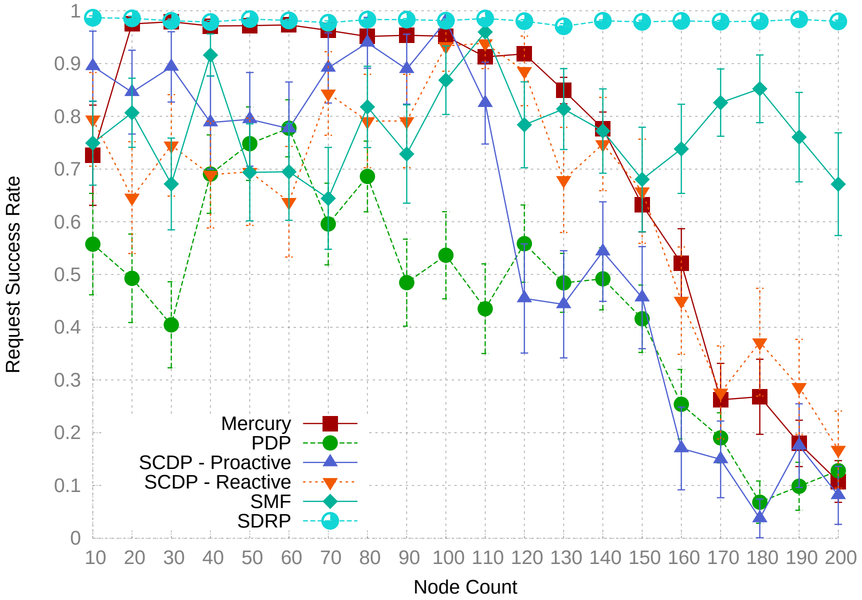

Fig. 4. Request Success Rate vs. Node Count. Medium contention and interference caused by control overheads have a marked effect on the success of service requests. SDRP's low control overheads and robust service advertisement approach maintain consistently high success rates.

3. C. Campo, C. García-Rubio, A. López, and F. Almenárez. Pdp: a lightweight discovery protocol for local-scope interactions in wireless ad hoc networks. *Computer Networks*, 50(17):3264–3283, 2006.
4. S. Corson. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. *IETF RFC 2501*, 1999.
5. J. Flathagen and K. Øvsthus. Service discovery using olsr and bloom filters. In *Proceedings of the 4th OLSR Interoperability Workshop*, October. 2008.
6. S. Giordano. *Mobile Ad Hoc Networks*, pages 325–346. John Wiley and Sons, Inc., 2002.
7. S. Gonzalez-Valenzuela, S. Vuong, and V. Leung. A mobile-directory approach to service discovery in wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 7(10):1242–1256, October. 2008.
8. J. Haerri, F. Filali, and C. Bonnet. Performance comparison of aodv and olsr in vanets urban environments under realistic mobility patterns. In *Proceedings of the 5th IFIP Mediterranean Ad-Hoc Networking Workshop*, pages 14–17, February. 2006.
9. R. Hekmat and P. Van Mieghem. Interference in wireless multi-hop ad-hoc networks and its effect on network capacity. *Wireless Networks*, 10:389–399, 2004.
10. T. Issariyakul and E. Hossain. *Introduction to Network Simulator NS2*. Springer Verlag, 2008.
11. P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings of the*

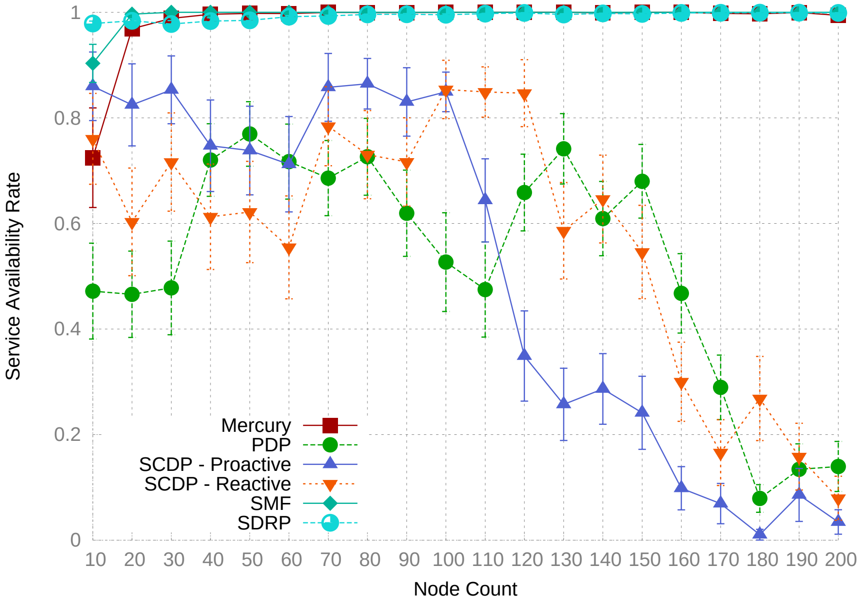

Fig. 5. Service Availability vs. Node Count. Here we can see that SDRP and cross-layer approaches maintain a high availability rate due to NS-2's prioritization of routing traffic and SDRP's lower control overheads.

IEEE Multi Topic Conference on Technology for the 21st Century, INMIC '01, pages 62–68, February. 2001.

12. L. Li and L. Lamont. A lightweight service discovery mechanism for mobile ad hoc pervasive environment using cross-layer design. In *Third IEEE International Conference on Pervasive Computing and Communications Workshops.*, pages 55–59, March. 2005.
13. C. Perkins, E. Belding-Rowyer, and S.Das. Ad hoc on-demand distance vector (aodv) routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999)*, pages 90–100, February 1999.
14. M. Rahman, F. Anwar, J. Naeem, and M. Abedin. A simulation-based performance comparison of routing protocols in mobile ad-hoc networks (proactive, reactive and hybrid). In *2010 International Conference on Computer and Communication Engineering (ICCCE)*, pages 1 –5, May. 2010.
15. V. Raychoudhury, J. Cao, W. Wu, Y. Lai, C. Chen, and J. Ma. Fast track article: K-directory community: Reliable service discovery in manet. *Pervasive and Mobile Computing*, 7(1):140–158, 2011.
16. R. Rivest. Rfc1321: The md5 message-digest algorithm. *RFC Editor United States*, 1992.
17. C. Toh, G. Guichal, D. Kim, and V. Li. Service location protocols for mobile wireless ad hoc networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):250–262, 2007.
18. A. Varshavsky, B. Reid, and E. de Lara. A cross-layer approach to service discovery and selection in manets. In *Proceedings of IEEE International Conference on*

Mobile Adhoc and Sensor Systems Conference, pages 466–474, November. 2005.

19. C. Ververidis and G. Polyzos. Routing layer support for service discovery in mobile ad hoc networks. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, 2005. PerCom 2005.*, pages 258–262. IEEE, 2005.
20. C. Ververidis and G. Polyzos. Service discovery for mobile ad hoc networks: A survey of issues and techniques. *Communications Surveys & Tutorials, IEEE*, 10(3):30–45, 2008.
21. X. Zhou, Y. Ge, X. Chen, Y. Jing, and W. Sun. Smf: A novel lightweight reliable service discovery approach in manet. In *7th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2011*, pages 1–5, September. 2011.