Concise Paper: LAAR: Long-range Radio Assisted Ad-Hoc Routing in MANETs

Ying Mao*, Jiayin Wang*, Bo Sheng* and Mooi Choo Chuah†
*Department of Computer Science, University of Massachusetts Boston, Email: {yingmao, jane, shengbo}@cs.umb.edu
†Department of Computer Science and Engineering, Lehigh University, Email:chuah@cse.lehigh.edu

***Abstract*—This paper investigates the routing protocol in smartphone-based mobile Ad-Hoc networks. We introduce a new dual radio communication model, where a long-range, low cost, and low rate radio is integrated into smartphones to assist regular radio interfaces such as WiFi and Bluetooth. We propose to use the long-range radio to carry out small management data packets to improve the routing protocols. Specifically, we develop new schemes built on the long-range radio to improve the efficiency of the path establishment process in the existing on-demand Ad-Hoc routing protocols. We have prototyped our solution LAAR on Android phones and evaluated the performance with small scale experiments and large scale simulation implemented on NS2. The results show that LAAR significantly improve the performance in terms of the overhead and the number of messages transferred in the network.**

I. Introduction

With the prevalence of smartphones, the scope of mobile applications has been significantly broadened in the past few years including almost every aspect in our daily life. This paper studies smartphone-based Ad-Hoc infrastructure to support applications that require interaction and communication in the proximity. It is motivated by the fact that location plays an extremely important role in mobile applications. A lot of location-based services for mobile phones have attracted a large volume of users [1]–[6]. We believe that there would be more attractive location-based applications developed if communication between nearby devices are well supported. For example, a police car at a crash site may disseminate the accident information to other cars within one mile distance; a student in library may send a picture via instant messenger to his friend in another classroom; and a user may share recently cached web pages with other nearby users who are requesting the same contents.

Constructing a mobile Ad-Hoc network (MANET) with hop-by-hop communication to carry local data traffic is desirable in practice. A MANET can effectively deliver data in the proximity without requiring the Internet connection. It is robust against infrastructure failure and can save unnecessary network bandwidth cost. However, the current routing protocols used in MANETs do not perform well in practice. One of the major issues is that it is costly and inefficient to establish a path from the sender to receiver. Traditional MANET routing protocols either pay a high cost for maintaining routing tables or flood a request message in the entire network for on-demand path discovery. Both categories require a large number of messages to be delivered for establishing a path. Another aspect of the inefficiency is the large overhead before the sender is able to start delivering data to the receiver, especially for sending a small amount of data. This paper particularly aims to improve the efficiency of the path establishment in on-demand Ad-Hoc routing.

We investigate a new *long-range radio assisted* Ad-Hoc communication model for smartphones as well as a suite of new techniques to significantly reduce the overhead of path establishment. We have prototyped this model on commercial Android phones by integrating additional long-range radio chips. Specifically, we adopt XE1205 [7] which features low cost (<\$30), low power ($10 \sim 20$mA current), and miles of communication range (1.6 miles for XE1205). However, as a tradeoff, its data rate is low (tens of bps) unsuitable for bulk data transmission. Therefore, we propose to use this additional radio channel for control and management messages to speed up the path establishment while data communication is still carried by WiFi or Bluetooth.

II. Related work

Routing in MANET has been an active area of research for years. Generally, there are two types of routing protocols in this area. One type of the routing is table driven protocols(proactive), such as DSDV [8], OSLR [9]. In these protocols, each node maintains one or more tables with routing information to every other node in the network. In this case, the protocols perform efficiently in path establishment. Whenever a route to a new destination is required, it has already existed at the source. However, they need to exchange a great number of messages to maintain the global routing table. The other type is on-demand protocol(reactive), such as DSR [10], AODV [11]. In on-demand protocols, the routes are created as required. In consequence, significantly fewer maintenances are needed. Considering MANET where routing table maintenance is usually costly, we mainly focus on the on-demand routing protocol design instead.

A lot of innovative approaches in this area have been studied to improve the network performance from different aspects. For example, in LQSR [12], the route is selected based on link quality metrics. Three performance metrics, i.e., expected transmission count (ETX), per-hop RTT, and per-hop packet pair, are implemented separately. However, LQSR fails to deal with topology changes. Frey et al. [13] focus on geographic routing to overcome topology changes impact. In [13], the

978-1-4799-6204-4/14 \$31.00 © 2014 IEEE
DOI 10.1109/ICNP.2014.57

node forwards packets by only using the position information of nodes in the network and the destination node. Another important aspect is to utilize multiple resources on one node to achieve better performance. For example, [14] attempts to use multi-channel on one node and proposes a hybrid channel assignment strategy–some interfaces on a node have a fixed assignment, while the rest can switch channels as needed. MR-LQSR [15], AODV-MR [16] and Extended-DSR [17] assume that each node is equipped with multiple radio interfaces. In the multi-radio environment, MR-LQSR uses a new metrics named weighted cumulative expected transmission time(WCETT) to provide better route selection via taking into account not only the link loss rate and bandwidth but also the interference among links that use the same spectrum channel as well as the channel diversity. The AODV-MR uses the multi-radio interfaces communication support to improve spectrum utilization and to reduce interference as well as contention in the network. Extended DSR attempts to address limited capacity and poor scalability problem by taking advantage of multi-radio feature. Zamree et al. [18] conduct a completed performance evaluation comparison across the previous three protocol.

Considering the utilization of multiple radio interfaces on one node, our project is closely related to [15]–[17]. In their settings, each node is equipped with two 802.11 wireless cards that introduce interference and channel allocation problem. Moreover, their emphasis is to improve the system stability by constructing multiple routes. However, in their protocol, path establishment process, the very first step, is still costly in terms of overhead and number of messages transferred. On the contrary, Our major focus is on addressing the initial step of every on-demand MANET routing protocol. Different from all previous work, we introduce the long-range radio on each node that operates at 915Mhz. By utilizing the long range feature, we largely reduce the overhead and number of messages transferred in the path establishment process.

III. System Model and Problem Formulation

In this paper, we consider a mobile Ad-Hoc network (MANET) consisting of smartphone nodes. Each smartphone is equipped with regular radios such as WiFi and Bluetooth, as well as the new long-range radio. We assume that the regular radio interface is configured to support Ad-Hoc communication between devices. Comparably, the other long-range radio has much longer communication range (up to 1.6 miles). However, the network bandwidth of the long-range radio is significantly lower than regular WiFi or Bluetooth radios. This paper targets at serving local network traffic where the sender and receiver are both in the MANET.

We adopt on-demand Ad-Hoc routing protocols such as DSR and AODV, where each node does not maintain stateful link information and a path is established only when the sender intends to transfer data to the receiver. Establishing a path from a sender to receiver is the very first and extremely important step in Ad-Hoc routing. In traditional MANET routing protocols, the basic idea is to flood route request messages (RREQ) initialized by the sender to the entire network until one of them reaches the receiver. Upon receiving the request, the receiver will send a route replay message (RREP) to the sender tracing back the transmission path of the RREQ message. Once the RREP is received, the sender will know the established path and the data packets will be delivered along it.

This flooding-based path establishment, however, is costly in terms of the time delay and number of messages transferred. First, RREQ message is blindly broadcast by a node to all its neighbors in omni-direction. Most of them are wasted and will never reach the receiver. Although RREQ message is often confined with a time-to-live (TTL) parameter, it still causes a large number of useless messages transferred which consume extra energy of each node and yield wireless signal interferences in the MANET. In addition, the exchange of RREQ and RREP takes a round-trip time with hop-by-hop delivery. Considering the interference and processing time at each relay node, this initial delay could be degrade the throughput performance especially when transferring a small amount of data.

In this paper, our goal is to use the long-range radio assisted mode to improve the performance of path establishment in on-demand Ad-Hoc routing protocols. Specifically, we aim to reduce the message flooded in the entire network and decrease the time overhead of establishing the path.

IV. Long-range Radio Assisted Ad-hoc Networks

In this section, we present our solution LAAR that takes advantages of the new long-range radio to improve the performance of path establishment. Our design is based on the following two techniques.

Bi-directional Message Flooding: In our hardware setting, the long-range radio could help the sender find receiver quickly. If the initial request is sent over the long-range radio it is highly likely that the receiver is within one-hop communication range. Based on this notification, we design a bi-directional path discovery protocol. Traditionally, the route request message (RREQ) is flooded from the sender towards the receiver. In our new model, once notified, the receiver could participate in this process as well. Therefore, both the sender and receiver can flood the RREQ message towards each other. When a node receives both request messages implying a path has been established, it can send the route reply to the sender through its long-range radio.

RSSI-guided Flooding: One of main disadvantages of flooding RREQ messages is the inefficiency. The messages are forwarded to all directions and most of them are wasted. In our solution, we confine the region involved in the flooding process by considering the received signal strength (RSSI) of the packets sent over the long-range radio. The basic intuition is that for any communication session, only the nodes "between" the sender and receiver should be involved. If a node is much further away from the receiver than the sender, its RSSI of a packet sent by the receiver should be lower than the sender's RSSI of the same packet.

Complete Protocol: Our solution LAAR includes a new three–way handshake protocol before flooding the RREQ messages. Assume source node S tries to send data to destination node D and they are within each other's communication range over the long–range radio. S first broadcasts an **INIT** message including S and D's IDs via the long–range radio. The combination of the source and destination's IDs $< S, D >$ uniquely identifies a communication session. Once receiving the message, D sends an **INIT-ACK** message back to S. Besides the source and destination's IDs, this message also includes the RSSI of the INIT message indicated by $R_{S \to D}$. Finally, S sends the last handshake message **INIT-FIN** including the RSSI of the INIT–ACK message ($R_{D \to S}$). Then, S starts to broadcast route request message (RREQ) towards D. Meanwhile, after receiving the INIT–FIN message, D will also send out RREQ towards S. At this point, the bi–directional RREQ flooding starts from both the source and destination.

In our solution, each node maintains a routing table to record the communication sessions it participates in as a relay node. Each row of the table represent a session including the source ID, destination ID, path to source, and path to destination. A node adds a new session in the routing table only when it receives all three handshake messages and the RSSI measurements indicate it is "between" the source and destination. Specifically, each node keeps another temporary table to record the candidate sessions that are still in the path establishment process. When overhearing INIT, INIT–ACK, and INIT–FIN messages, a node applies the following protocol: (1) When receiving an INIT message, the node adds a new entry into the temporary table recording the source and destination of the new session as well as the RSSI of this message indicated by R_S. (2) When receiving an INIT–ACK message, the node first checks its temporary records and looks for the matching session information. If the session (<src, dst> pair in the INIT–ACK) cannot be found, the node will discard the message. If the session is found, i.e., the node has received the corresponding INIT message, the node will check the recorded RSSI of the INIT (R_S) and compare to the RSSI value ($R_{S \to D}$) in the INIT–ACK message. The new session will be removed from the temporary records if $R_S < \beta \cdot R_{S \to D}$ where $\beta \in (0, 1)$ is a threshold depending on the signal propagation model. This step filters out the nodes that are further away from the source node than the destination. If $R_S \geq \beta \cdot R_{S \to D}$, the node will add the RSSI of this INIT–ACK message indicated by R_D into the record. (3) When receiving an INIT–FIN message, similar to the second step, a node checks the temporary table and searches the matching session information. If the session has been recorded, the node reads R_D from the record and compares it to the value of $R_{D \to S}$ in the INIT–FIN message. If $R_D < \beta \cdot R_{D \to S}$, the node will eliminate the session from the temporary table.

After the three–way handshake protocol, the RREQ message will be flooded from both the source and destination. In our solution, a RREQ message includes source/destination IDs, the nodes it has traversed (i.e., the path), and an additional field indicating the direction of the message, i.e., from the source node or destination node. Having received a RREQ, each node checks its temporary table. If the session of RREQ exists in the temporary table, the node will add its own in the field of path and further broadcast the RREQ. Otherwise, the RREQ message will be discarded. Once a node receives the RREQ messages for the same session from both sides, it will broadcast an announcement message (**ANNO**) via the long–range radio with a complete path from the source to destination. After receiving the ANNO message, every node will no longer forward the RREQ message for this session. In addition, each node checks the path and add the session to its routing table if it is listed in the path.

INIT	src	dst			
INIT–ACK	src	dst	$R_{S \to D}$		
INIT–FIN	src	dst	$R_{D \to S}$		
RREQ	src	dst	TTL	path ($S \to A \to$ ...)	dir
ANNO	src	dst	path ($S \to A \to$ ... $\to B \to D$)		

TABLE I: Message Format

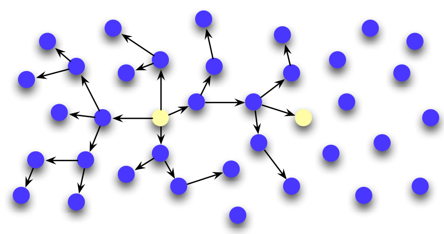

Fig. 1: Traditional Path Establishment (TTL=4)

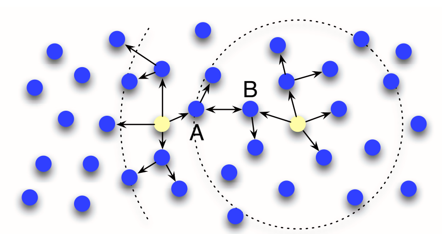

Fig. 2: Path Establishment in LAAR

Fig. 1 and Fig. 2 show a comparison between traditional path discovery and our long–range radio assisted path discovery. The two orange nodes are sender S and receiver D. Fig. 1 shows the request message flooding with TTL (time to live) set to 4. The shortest path from sender to receiver is 3–hop long and in this partial topology, 14 nodes broadcast the request when it reaches the receiver. Fig. 2 illustrates the benefits of bi–directional flooding and RSSI filtering. In this example, the request is propagated from both sender and receiver and the path is established in the second round of broadcast, i.e., when node A and B broadcast their received requests. With the handshake messages including RSSI information, we assume the dotted circle and arc centered at the receiver define the region where the RSSI of the receiver's packets is similar, i.e., $R_{D \to S}$. Assume the nodes on the left side of the dotted arc have RSSIs (R_D) smaller than $\beta \cdot R_{D \to S}$. Thus they will not forward the RREQ message. Only 7 nodes broadcast the RREQ message in Fig. 2 when the path is established.

V. System Implementation

Designing the new dual-radio communication model is motivated by the capabilities of smartphones serving as mobile nodes in MANETs. The current smartphone hardware architecture and operating system have been well developed providing flexible interfaces for designers to integrate new hardware. In this paper, we attach the long-range radio transceiver, Xemics XE1205 [7], to smartphones as the first attempt of implementing dual radio model for smartphone Ad-Hoc networks .

Implementing the dual radio model on smartphones is challenging in both hardware connection and software support. In our work, we have successfully attached an external long-range radio to smartphones and implemented basic communication modules. Our prototypes adopts TinyNode [19] as the external device consisting of an XE1205 [7]. It operates on 915Mhz and feature low cost, low power consumption, and miles of communication range (1.6 miles for XE1205). However, the bandwidth is extremely low ranging from several Kbps to tens of Kbps depending on the working mode.

We have integrated the long-range radio into assorted phones including HTC Magic phone, Nexus One phone, and Nexus 4 phone. We use a USB to serial port converter which equips with PL2303 [20], a low cost and high performance USB-to-Serial bridge controller, to connect TinyNode and smartphone (through either ExtUSB or MicroUSB port).

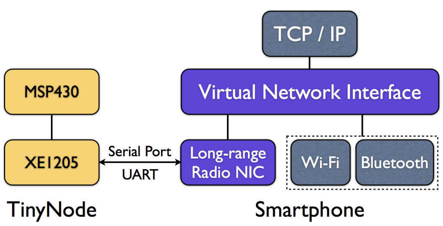

Fig. 3: Software Architecture

Software support includes programs on both smartphones and the external devices. Fig. 3 illustrates the design architecture with TinyNode. We have customized Android kernel and developed user space programs on smartphones to support dual radio communication. Basically, the USB port of a phone is recognized as a serial UART device (Universal Asynchronous Receiver/Transmitter) and a device file for it is created under '/dev/'. User programs can communicate with the USB port by reading from or writing to the new device file. Communication between a TinyNode and smartphone is built on a module deployed on both sides. We have implemented data-link level protocol over this serial link (UART) communication including basic mechanisms such as checksum and retransmission. In addition, we use TUN/TAP device driver [21] to create a virtual network interface and change the routing policy on phones such that all incoming and outgoing traffic will pass through the virtual interface. Specifically, TUN simulates a network layer(layer 3) device and processes layer 3 IP packets. TAP simulates a link layer(layer 2) device and processes layer 2 packets such as Ethernet frames. In our solution, TUN is used with routing, while TAP is used for creating a network bridge. Then we write programs in TUN/TAP driver to process each packet. Our prototype smartphone is able to dispatch each packet to different network interfaces, either WiFi, Bluetooth, or the long-range radio.

Fig. 4 shows two prototype smartphones equipped with TinyNode conducting a ping test with dual radio model. First, two smartphones are connected with WiFi Direct interface. Then, one smartphone(sender) sends ping-request messages to the other(receiver) through TinyNode interface and, upon receiving ping-request packets, the receiver sends back ping-reply messages through WiFi Direct interface. The sender displays round trip time of a ping-request and tcpdump is running on the receiver side to monitor the behavior of its virtual network interface.

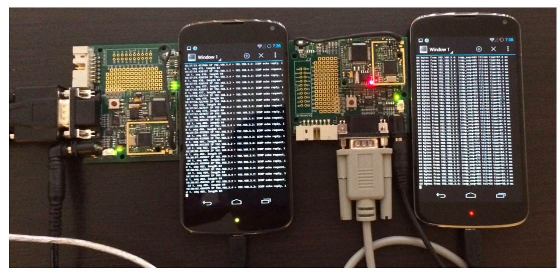

Fig. 4: Demonstration of Dual Radio Model

VI. Performance Evaluation

In this section, we evaluate LAAR and compare it with the conventional MANET routing protocols. The results are drawn from experiments on basis of a small scale network and NS2 [22] simulation on basis of a large scale network. Our major performance metrics are the overhead and the number of messages transferred during the process of path establishment.

We compare LAAR with DSR [10], DSR-R0 (default implementation of DSR in NS2), and AODV-ERS [23]. DSR-R0 is a variant version of DSR with *ring-zero search* in the process of path establishment. Ring-zero search aims to reduce the overhead by firstly sending an RREQ with TTL=0. If the sender and receiver are direct neighbors, the path would be quickly established. Otherwise, upon a timer expires, the sender will send another RREQ with a regular TTL value. AODV-ERS is an enhanced version of AODV [11] with expanding ring search, where the sender broadcasts the RREQ for multiple rounds each with an incremental TTL value. The process terminates when the destination is reached.

In our evaluation, we set the RREQ messages' maximum TTL to 5 in LAAR, DSR and DSR-R0. For AODV-ERS, we set the initial TTL to 2, and the expanding value of TTL is set to 2, i.e., TTL is increased by 2 every round. In addition, for all protocols, the random backoff window for broadcasting, which is used as a backoff to avoid congestion in the network, is set to be 10ms and timeout (in DSR-R0 and AODV-ERS) for RREQ is set to be 30ms.

A. Experimental Results

First, we build a small scale Ad–Hoc network consisting of 6 Android smartphones positioned in a straight line. Since the Ad–Hoc mode is not supported in the current protocols, we develop a module on each phone to relay data by bridging two network interfaces, WiFi and WiFi–Direct, i.e., one interface connecting to the uplink neighbor and the other connecting to the downlink neighbor. In the experiments, the hop distance between the sender and receiver ranges from 1 to 5. Fig. 5 shows the results of path establishment. The time overhead grows almost linearly with the hop distance in LAAR. Apparently, our solution outperforms the other protocols. With a 5–hop path, LAAR reduces the overhead by 45% compared to DSR which yields the second best performance.

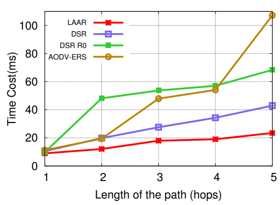

Fig. 5: Overhead of Path Establishment

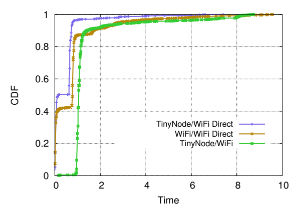

Fig. 6: Overhead of Switching between Interfaces

Fig. 6 illustrates the breakdown processing delay on each node caused by switching between network interfaces. According to our design, there are three kinds of switching between interfaces, TinyNode and WiFi, WiFi and WiFi Direct, and TinyNode and WiFi Direct. As shown in Fig. 6, when considering 90% of the packets, the delays are 1.722ms, 1.281ms, and 0.746ms for switching between WiFi and WiFi Direct, WiFi and TinyNode, and TinyNode and WiFi Direct, respectively. Switching between WiFi and WiFi Direct is more costly, because these two interfaces share the same hardware.

B. Simulation

In addition, we conduct simulation to evaluate LAAR in a large scale network.

1) Simulation Settings: We consider the following two topology settings. (1) **Grid topology:** In grid topology, each node has two(node at edge) or four neighbors and the distance between any two neighboring nodes is identical. (2) **Random topology:** In random topology, nodes are randomly placed in a square field. The workload we consider includes single communication session and multiple communication sessions.

We tune the parameters in NS2 to facilitate our network settings. First, for wireless signal prorogations, we adopt two–ray ground reflection model and constant speed propagation delay model. In addition, each node in our LAAR protocol is set with two radios. We modify the NS2 to support two wireless interfaces. The frequency of the long–range radio is set to 915MHz, and the communication range is configured to be 2500m in receiving (RX) and 3000m in carrier sensing (CS). The other regular radio (short range) is configured to work at 2.4GHz, and the RX and CS ranges are set to be 100m and 500m respectively.

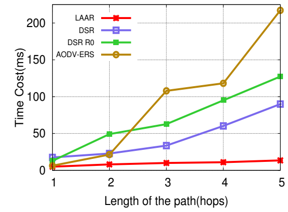

Fig. 7: Overhead (single session)

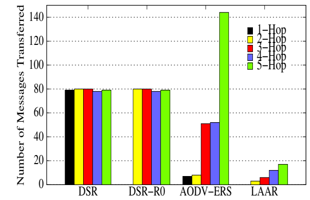

Fig. 8: Number of messages transferred

2) Grid Topology: Fig. 7 shows the overhead results of single session with a 700m×700m grid topology. We choose sender/receiver in the cental region and enumerate their distance from 1 hop to 5 hops. According to Fig. 7, the overhead of path establishment process in DSR, DSR–R0 and AODV–ERS increases substantially along with the distance between the sender and receiver. Our solution, LAAR, however, shows a much flatter curve and significantly reduces the overhead with the help of the long–range radio. For example, to construct a 5–hop path, LAAR costs 13.453ms, while DSR, DSR–R0 and AODV–ERS spend 89.947ms, 127.351ms and 217.715ms respectively. The improvement of the overhead is mainly due to the bidirectional flooding and the instant announcement of the established path over the long–range radio.

Additionally, we present the total number of messages (RREQs and RREPs) transferred in the path establishment process in Fig. 8. The performance of DSR is almost constant regardless of the distance between the sender and receiver. The flooding is only controlled by a fixed TTL value and more than half nodes in the field are involved in the process. DSR–R0 has a similar performance except for the case when the sender and receiver are direct neighbors (1–hop distance). AODV–ERS performs well only when the hop distance between the sender and receiver is small. For 5–Hop case, the number of messages transferred in AODV–ERS is almost double the number in DSR. Our solution, LAAR, dramatically reduces the number of messages transferred in the path establishment process. For example, only 12 messages transferred in constructing a 5–hop route, compared with 79, 79 and 144 for DSR, DSR–R0 and AODV–ERS.

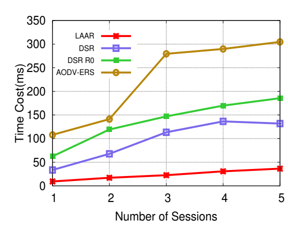

Fig. 9: Overhead (multiple sessions)

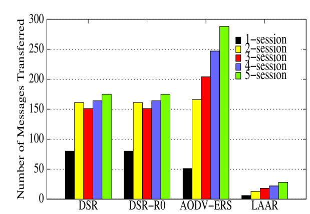

Fig. 10: Number of messages transferred

Furthermore, we evaluate the performance of LAAR with multiple concurrent sessions. We choose 1 to 5 different pairs of sender and receiver that are 3–hop apart in a 700m×700m grid topology. Fig. 9 shows the average value of the overhead. Again, LAAR is greatly superior to all other schemes.

Since the number of messages transferred in LAAR is better controlled, the interference and congestion brought by multiple communication sessions have a limited impact on the overhead performance. Fig. 10 depicts the number of messages transferred with multiple sessions. The performance of DSR, DSR-R0, and AODV-ERS is quite similar while LAAR reduces the number of messages by a factor of six. For example, when there are 5 concurrent 3-hop sessions, LAAR only incurs 28 messages to establish 5 paths and the other three protocols transfer 175, 175 and 380 messages.

3) Random Topology: In the random topology, we place 300 nodes into a 600m×600m topology. Specifically, we generate 300 different topologies and randomly pick the sender and receiver. Then we collect the results based on the same hop distance between sender and receiver to calculate the average overhead and number of messages transferred.

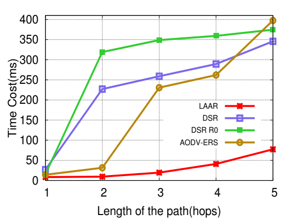

Fig. 11: Overhead (single session)

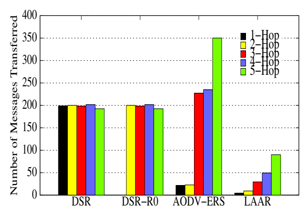

Fig. 12: Number of messages transferred

The overhead performance is displayed in Fig. 11. Compared to the grid topology, the overhead here increases for the same hop distance setting. For instance, time cost of 2-hop route of DSR in grid and random topology is 49.230ms and 227.072ms. This significant increase is caused by the density of the topology. In the grid topology, each node only has up to 4 neighbor nodes. However, in our 300 random topologies, the average number of neighbors is 24.325. The dense topology can lead to much higher congestion. However, in LAAR, the overhead of path establishment remains low, since the two-radio system helps reduce the number of messages transferred mitigating the congestion's effect. The number of messages transferred in random settings (Fig. 12) shows a similar trend in all the protocols, but the number increased due to the density of the topology. LAAR, again, outperforms the other three tested protocols.

We apply the same 300 random topologies to conduct the experiments with multiple sessions. The path length of each session is chosen to be 3-hop. Fig. 13 shows the average overhead, among which the performance of DSR, DSR-R0 and AODV-ERS remain at high level (above 250ms). LAAR, on the contrary, achieves low cost(less than 100ms) which benefits from long-range radio. Fig. 14 presents the number of messages transferred for constructing a path. From 1 to 2 sessions, the time cost experiences a large increase for DSR, DSR-R0 and AODV-ERS, which is caused by two RREQs in the path establishment process. Again, LAAR remains at a very low cost level. For example, from 1 to 2 sessions, LAAR only requires 22 and 34 messages transferred to discover the path.

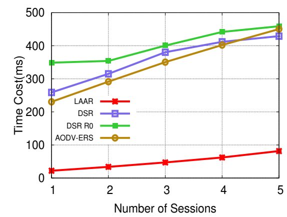

Fig. 13: Overhead (multiple sessions)

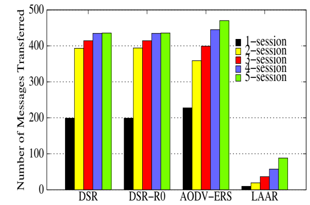

Fig. 14: Number of messages transferred

VII. Conclusion

This paper presents LAAR, a new dual radio model for smartphone-based Ad-Hoc networks. We integrate a long-range radio to help improve the performance of path establishment which is a critical component in the existing routing protocols. The experimental and simulation results show that LAAR dramatically reduces the overhead and the number of messages transferred in the network.

References

[1] Foursquare. http://www.foursquare.com.
[2] Facebook Places. http://www.facebook.com.
[3] Yelp. http://www.yelp.com.
[4] Waze. http://www.waze.com.
[5] SCVNGR. http://www.scvngr.com.
[6] UBER. http://www.uber.com.
[7] Xe1205. http://www.semtech.com/images/datasheet/xe1205.pdf/.
[8] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, SIGCOMM '94, pages 234–244, 1994.
[9] Pl2303 android solution. http://www.ietf.org/rfc/rfc3626.txt.
[10] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181, 1996.
[11] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *THE 2ND IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, pages 90–100, 1997.
[12] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. *SIGCOMM Comput. Commun. Rev.*, 34(4):133–144, August 2004.
[13] Hannes Frey. Scalable geographic routing algorithms for wireless ad hoc networks. *Network, IEEE*, 18(4):18–22, 2004.
[14] Pradeep Kyasanur and Nitin H. Vaidya. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 10(1):31–43, January 2006.
[15] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 114–128, 2004.
[16] Asad Amir Pirzada, Ryan Wishart, and Marius Portmann. Multi-linked aodv routing protocol for wireless mesh networks. In *GLOBECOM*, pages 4925–4930. IEEE, 2007.
[17] Saad Biaz, Bing Qi, Shaoen Wu, and Yiming Ji. In *Evaluation of Multi-Radio Extensions to DSR for Wireless Multi-Hop Networks*, pages 65–69.
[18] Aisha Hassan Abdalla Zamree Che-aron and Khaizuran Abdullah. The performance evaluation of aodv-based and dsr-based multi-radio routing protocols in cognitive radio ad hoc network. *Research Journal of Applied Sciences, Engineering and Technology*, 10(1), December 2012.
[19] Tinynode 584. http://www.tinynode.com/?q=system/files/TN584_Users_Manual_v_1_3.pdf.
[20] Pl2303 android solution. http://prolificusa.com/files/PL2303%20Android%20brochure%2020121101.pdf.
[21] Tun/tap. http://en.wikipedia.org/wiki/TUN/TAP.
[22] Network simulator 2. http://www.isi.edu/nsnam/ns/.
[23] Woonkang Heo and Minseok Oh. In *FGCN (2)*, pages 128–132.