Building Smartphone Ad-Hoc Networks With Long-range Radios

Ying Mao Jiayin Wang Bo Sheng
Department of Computer Science, University of Massachusetts Boston
{yingmao, jane, shengbo}@cs.umb.edu

Abstract—**This paper investigates the routing protocols 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 to improve the efficiency of the path establishment and path recovery process in the 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 improves the performance.**

I. INTRODUCTION

This paper studies smartphone-based Ad-Hoc networks to support applications that require interactions and communications 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]. The current location-based services, however, are still built upon the client-server architecture which incurs some unavoidable issues. We believe the Ad-Hoc network can certainly help improve this category of applications. Let us consider an example where a department store in a mall tries to deliver a flyer file to a nearby shopper. In the current architecture, the following steps are required: (1) the store hosts the file on its server; (2) the user has to install the store's app; (3) the user connects to the Internet in the mall and reports his location to the store's server; (4) the server delivers the flyer file to the user. This process involves a few representative drawbacks that an Ad-Hoc network can help address. First, a user cannot discover nearby data or information (step 2). He has to register for each and every service he is interested in. With an Ad-Hoc network, a user can browse or receive all the unknown services nearby as long as they transfer information on a common channel (such as WiFi). Second, step 3 and step 4 require the Internet connection which may not be always available, e.g., subway stations, crowded and congested areas, and the areas with infrastructure failures. Not to mention that when the store and user are close to each other, the data transfer going through the Internet may not be necessary and could incur additional costs to the user. Third, step 3 requires an accurate indoor localization scheme. With an Ad-Hoc network, hearing the signals from the store is the best evidence that the user is close to the store. Therefore, constructing a mobile Ad-Hoc network (MANET) with hop-by-hop communication to carry local data traffic is desirable in practice. However, the current routing protocols used in MANETs suffer from two major problems. First, it is costly and inefficient to establish a path from the source to destination. 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. The second problem in the current MANETs techniques resides in the path recovery protocol which is a critical component because the established paths are often broken in highly mobile scenarios. The current path recovery is usually triggered by an observed failure and proceeds by repeating the path establishment process. In practice, however, this recovery process is slow.

In this paper, we investigate a new *long-range radio assisted* Ad-Hoc communication model for smartphones as well as a suite of new techniques to significantly improve the performance of path establishment and recovery. 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 $\sim$ 20mA current), and a communication range of 1.6 miles. However, as a tradeoff, its data rate is low (tens of bps) unsuitable for bulk data transmission. We propose to use this additional radio channel for control and management messages while data communication is still carried by WiFi or Bluetooth.

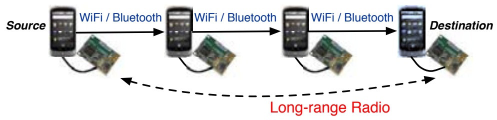


Fig. 1: Long-range Radio Assisted Model

The major contributions of this paper are: (1) We introduce a new dual-radio model with a long-range radio interface for smartphones. (2) We develop a new Ad-Hoc routing protocol with the assistance of the long-range radio. Our protocol has enhanced the path establishment and recovery protocols. (3) We have evaluated our solution with small scale experiments and large scale simulation on NS2. The results show a significant improvement on the performance.

II. RELATED WORK

Generally, there are two types of routing protocols in MANETs. One type is proactive protocols such as DSDV [8],

978-1-4673-8590-9/15/$31.00 ©2015 IEEE

OSLR [9]. In these protocols, each node maintains one or more tables with routing information to every other node in the network. 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 this paper, we mainly focus on the on-demand routing protocol design.

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, such as expected transmission count (ETX), per-hop RTT, and per-hop packet pair. However, LQSR only works with static setting and fails to deal with topology changes. Frey et al. [13] focus on geographic routing to overcome topology changes with mobility. 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. Multiple resources on one device make the routing protocols more flexible. MR-LQSR [15], AODV-MR [16] and Extended-DSR [17] assumes that each node is equipped with multiple radio interfaces. MR-LQSR uses a new metric named weighted cumulative expected transmission time(WCETT) to provide better route selection. The AODV-MR uses the multi-radio interfaces communication to improve spectrum utilization and reduce interference. Extended-DSR attempts to address limited capacity and poor scalability problem by taking advantage of multi-radio feature.

Our work is closely related to [15]–[20]. However, in their settings, each node is equipped with multiple 802.11 wireless cards. And they focus on addressing issues of interference and channel allocation. In our case, the two interfaces work on different frequencies. We mainly focus on utilizing the collaborations of these radios to boost the network performance. In our previous work [21], we investigate the route construction and as a following work, we design the whole protocol.

III. BACKGROUND AND PROBLEM

We target on the routing problem in a MANET where a source node aims to transfer data to a destination node. Each node in our setting is a smartphone and the phone-to-phone Ad-Hoc communication is carried out by WiFi or Bluetooth interface. 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 source intends to transfer data to the destination. Compared to proactice protocols, on-demand routing protocols are more suitable in a dynamic network considering user mobility. Here we introduce the details of some key components in the traditional on-demand Ad-Hoc routing protocols and list their drawbacks which motivate our new design.

Path Establishment: Establishing a path from the source to destination is a basic and important step in Ad-Hoc routing. The basic design in the prior work is to let the source flood a *route request message* (RREQ) to the entire network until one of them reaches the destination. Then the destination will send a *route replay message* (RREP) to the source tracing back the transmission path of the RREQ. Once the RREP is received by the source, a routing path is successfully established. This flooding-based scheme, however, is costly in two aspects. First, the RREQ message is broadcast by a node to all its neighbors in omni-direction. Most of the messages will never reach the destination. 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 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 degrade the throughput performance especially when transferring small amounts of data.

Path Cache: Path cache is often included in the MANET routing protocols to avoid unnecessary path establishments. Each node stores the paths from itself to other nodes into a *route cache* based on the RREQs or RREPs it overhears. In the path establishment, when an RREQ arrives, the node will first check its route cache. If there exists a path to the destination in the route cache, the node will reply to the source without propagating the RREQ. In this scheme, the validity of the cached paths is crucial to the performance. Because of the node mobility, the cached paths may not be available when the node intends to use them. In the traditional routing protocols, the validity of the cached paths is never checked. When a node attempts to use a cached path and later finds the path is stale, the source will have to re-establish another path.

Path Recovery: Path recovery is another important component in MANET routing protocols. When a node on the path moves out of the transmission range of its neighboring hops, a link is broken and the path recovery protocol will be triggered to establish another path. In typical MANET routing protocols, the node that fails to receive a link layer acknowledge detects the broken link and sends a *route error message* (RERR) back to the source. The source will first search its route cache for an alternative route to the destination. If no alternative is found, the source initializes a new path establishment process. In addition, any node that receives or overheads an RERR message will delete all the routes in the cache that contain the broken link. Path recovery inherits the issues we have mentioned for path establishment and path cache.

Our Problem Setting: In this paper, we aim to develop an efficient MANET routing protocol based on the integration of a long-range radio interface that helps address the above issues. Specifically, we consider a MANET consisting of smartphone nodes and each smartphone is equipped with two heterogeneous wireless interfaces: one is the regular wireless radios such as WiFi and Bluetooth, and the other is a new long-range radio. According to our prototype, the long-range radio has a much longer communication range (up to miles) than WiFi or Bluetooth. Its power consumption is extremely low which is suitable for smartphones. However, the network bandwidth of the long-range radio is significantly lower than the regular radios. More details of the hardware characteristics will be introduced in Section V. In our solution, the long-range radio will be used to broadcast small management packets while the data transfer is still carried by the regular radios in a hop-by-hop fashion. In addition, considering the coverage of the long-range radio, this paper targets at the local data transfer where the source and destination can directly reach each other over the long-range radio. Our goal in this paper is to use the new long-range radio assisted communication model to improve the performance of MANET routing protocols. Specifically, with the new long-range radio interface, we aim to reduce the message flooded in the entire network, decrease the time overhead of establishing or recovering a path.

IV. SYSTEM DESIGN OF LAAR

In this section, we present our solution LAAR which mainly consists of an efficient path establishment protocol and path recovery protocol. In addition, we develop a new route cache management scheme that serves both path establishment and recovery protocols.

A. Path Establishment

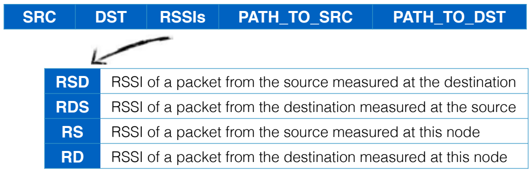
1) Motivations: Our design of the path establishment process includes the following two major new techniques.

Bi-directional Route Request Flooding: Traditionally, the route request message (RREQ) is flooded from the source towards the destination. In our problem setting, the source and destination are directly connected over the long-range radio. Thus, before flooding the RREQ message, the source can use the long-range radio to notify the destination about the upcoming communication session. An then, the destination will participate in this process as well. In out solution, therefore, both the source and destination flood the RREQ message towards each other. When a node receives both request messages implying a path has been established, it can send an announcement message through its long-range radio.

RSSI-guided Flooding: Traditionally, the RREQ 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 source and destination should be involved in propagating the RREQ messages. For example, if a node is further away from the destination than the source, it should not forward the RREQ for the session. Our solution provides relevant RSSI information to each node to help determine if the node should participate in the flooding process.

2) Complete Path Establishment Protocol:: Our path establishment protocol includes three phases. In the first phase, we develop a new *three-way handshake protocol* to enable the bi-directional route request flooding and prepare for a *RSSI-guided flooding*. In the second phase, the source and destination each broadcasts an RREQ and all the participating nodes propagate the RREQs as in the traditional routing protocols. The third phase is for announcing an established path. We assume that each node maintains a regular routing table that hosts the routing information for all the active communication session it participates. The routing table contains at least three columns: the source node ID (SRC), the destination node ID (DST), and an ordered list of node IDs that represent the path to the destination (PATH_TO_DST). After a path is successfully created, each node in the path will add a new entry in its routing table for this session.

In our solution, each node keeps an additional table, called *preparation table*, to help decide whether the node will participate in a path establishment process. The structure of a preparation table is illustrated in Fig. 2. Compared to the routing table, each entry in the preparation table includes some more fields such as PATH_TO_SRC and four RSSI values.



SRC	DST	RSSIs	PATH_TO_SRC	PATH_TO_DST

RSD	RSSI of a packet from the source measured at the destination
RDS	RSSI of a packet from the destination measured at the source
RS	RSSI of a packet from the source measured at this node
RD	RSSI of a packet from the destination measured at this node

Fig. 2: Preparation table structure

Next, we present the details of the three phases where the preparation tables will assist to eventually update the routing tables on the path from the source to destination.

Phase I - Three-way handshake: Assume the source S tries to send data to the destination D and they are within each other's communication range over the long-range radio. S first sends out 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 ($R_{D \to S}$). The structures of these three messages are illustrated in Fig. 3.

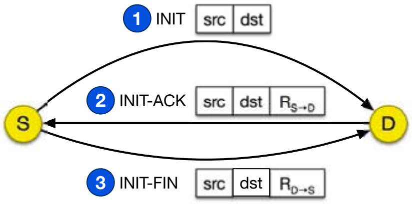


Fig. 3: Three-way handshake protocol

When overhearing the three-way handshake messages, every node that is neither source or destination applies the following Algorithm 1. Basically, the preparation table hold the candidate sessions that may be added into the routing table later. When a node receives an INIT message (lines 3–4), it adds a new entry into the preparation table recording the new session as well as the RSSI of this message (RS). INIT-ACK and INIT-FIN messages confirm the awareness of the incoming path establishment process at the source and destination, and also help enforce the RSSI-guided flooding. When receiving an INIT-ACK message (lines 5–11), the node first searches its preparation table for the matching session with <src, dst>. If a matching entry E is found, the node will compare the recorded RS value with the RSSI value ($R_{S\to D}$) in the INIT-ACK message. This entry E will be removed from the preparation table if E.RS$<\beta\cdot R_{S\to D}$ where $\beta\in(0,1)$ is a threshold depending on the signal prorogation model. This step filters out the nodes that are further away from the source node than the destination. If E.RS$\geq\beta\cdot R_{S\to D}$, the node will update the entry E by setting E.RD value to be the RSSI of this INIT-ACK message. Similar steps are applied when processing an INIT-FIN message (lines 12–18).

Algorithm 1 Process Three-way Handshake Messages

```
 1: function Receive(msg):
 2: Read msg.src, msg.dst, and measure the RSSI of the message
    indicated as msg.rssi
 3: if msg is an INIT message then
 4:     Add a new entry {SRC=msg.src, DST=msg.dst, RS=msg.rssi}
        into the preparation table
 5: else if msg is an INIT-ACK message then
 6:     Search the session <msg.src, msg.dst> in the preparation table
 7:     if there exists an entry E for the session then
 8:         if E.RS< $\beta\cdot$ msg.$R_{S\to D}$ then
 9:             Remove this entry E from the preparation table
10:         else
11:             E.RSD=msg.$R_{S\to D}$ and E.RD=msg.rssi
12: else if msg is an INIT-FIN message then
13:     Search the session <msg.src, msg.dst> in the preparation table
14:     if there exists an entry E for the session then
15:         if E.RSD value is null or E.RD< $\beta\cdot$ msg.$R_{D\to S}$ then
16:             Remove this entry E from the preparation table
17:         else
18:             Update the entry by setting E.RDS=msg.$R_{D\to S}$
```

Phase II - Bi-directional RREQ Flooding: In phase II, both the source and destination will start flooding an RREQ message towards each other. Essentially, a node will participate in flooding an RREQ only if its preparation table contains an entry for the session of the received RREQ. In our solution, an RREQ message includes source/destination IDs, a TTL value, the nodes it has traversed (i.e., the path), and an additional field indicating the origin of the message, i.e., from the source node or destination node. The following Fig. 4 illustrates the message structure. Having received an RREQ, each node applies the following Algorithm 2. First, it searches its preparation table for the session this RREQ represents. If there exists an entry for the session, the node will add its own ID in the field of the path and further broadcast the RREQ if the TTL is not expired. Meanwhile, the node will add the path included in the RREQ into either E.PATH_TO_SRC or E.PATH_TO_DST according to the origin of the RREQ. If the node finds that both E.PATH_TO_SRC and E.PATH_TO_DST have been filled, it will move to Phase III to announce the established path.

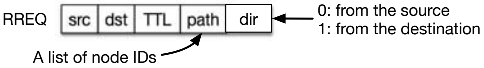


Fig. 4: RREQ message format

Phase III: Path Announcement 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. An ANNO message contains only three fields: the source (src), the destination (dst), and the full path from the source to destination. After receiving the ANNO message, every node will no longer forward the RREQ message for this session (by removing the entry for the session in the preparation table). In addition, each node checks the path and adds the session to its routing table if it is listed in the path.

Algorithm 2 Process RREQ Messages

```
function Receive(msg):
Read msg.src and msg.dst, and search the preparation table
if there exists an entry E for the session then
    if msg.dir indicates msg is from the source then
        if E.PATH_TO_SRC = null then
            E.PATH_TO_SRC = msg.path
    else
        if E.PATH_TO_DST = null then
            E.PATH_TO_DST = msg.path
    if E.PATH_TO_SRC and E.PATH_TO_DST are defined then
        SendAnnouncement(E)
    else if msg.TTL>0 then
        Broadcast a new RREQ {msg.src, msg.dst, msg.TTL-1,
        msg.path+NodeID, msg.dir}
```

B. Path Recovery

Path recovery is a critical component in MANETs because of the dynamic network topology caused by user mobility. We develop an efficient path recovery protocol in LAAR with the following two new techniques. Due to the page limit, we omit the detailed pseudo codes for the protocols.

Partial Path Recovery: In the prior work, once a node detects a broken link, the notification will be sent back to the source by an RERR message, and then the source will launch a new path establishment process. Therefore, any single link failure will lead to a complete path establishment process which is not efficient in practice, especially if the broken link is shared by multiple active sessions. An example is shown in Fig. 5. In the traditional MANET routing protocols, while node V moves away causing a broken link, node U will send three RERRs to the sources which will further start three path establishment processes.

In our partial path recovery solution, the node who detects the failure will notify the sources with a single RERR over the long-range radio and start path establishment processes with the destinations. Referring to the example in Fig. 5, node U will attempt recover the paths from U to D1 and D2. If successful, node U will notify the sources about the recovered paths over the long-range radio. Meanwhile, each source node also sets a timer once receiving an RERR. If the detecting node cannot recover the path to the destinations before the timers expire, the sources will initialize the path establishment process with the destinations.

Proactive Path Recovery: The other new technique we develop is to proactively start path recovery protocol before any link is broken. The basic idea is to detect weak or about-

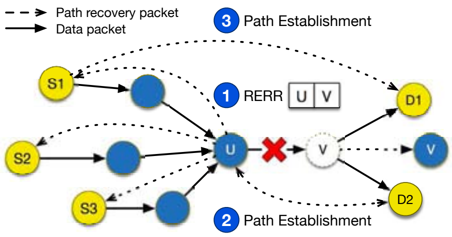


Fig. 5: An example of partial recovery: 3 active sessions (S1,D1), (S2,D2), and (S3,D2) share a link U→V. Node V moves away and the link U→V is broken.

to-break links based on each phone's mobility. Considering smartphones being the mobile nodes in our setting, we particularly use the accelerometer and RSSI measurements to determine if a node is moving away from a path it belongs to. If a node detects high movements or poor RSSIs from its neighbors, it will notify the neighboring nodes about the possible departure. Then a path recovery process will start when the node still carries out the data transfer. Once a new path is established, the neighboring nodes will update their routing tables to bypass the departing node.

Algorithm 3 Proactive Path Recovery

```
1:  $RL$: avg RSSI level, $SL$: avg speed level, $C = 0$
2:  When a new packet is received, update $R$
3:  if $RL$ is GOOD then
4:      $C = 0$; return;
5:  else if $RL$ is POOR then
6:      $C = 0$; Start the recovery process;
7:  else
8:      Measure the user's moving speed $SL$
9:      $C = \begin{cases} C + \Delta_H & \text{: if } SL \text{ is HIGH} \\ C + \Delta_M & \text{: if } SL \text{ is MEDIUM} \\ C + \Delta_L & \text{: if } SL \text{ is LOW} \end{cases}$
10:     if $C \geq \tau$ then
11:         $C = 0$; Start the recovery process;
```

In practice, both accelerometer and RSSI readings are dynamic, and may not accurately reflect the user movements. We develop a heuristic algorithm that combines these two measurements to indicate if a node will cause a broken link soon. Algorithm 3 illustrates the detailed process when a packet arrives. Specifically, we define three discrete levels for RSSI values, $\{$*GOOD, FAIR, POOR*$\}$. While a *GOOD* RSSI indicates a stable link, a *POOR* RSSI will trigger a path recovery process. When a *FAIR* RSSI is received, our solution will start to periodically measure the accelerometer. Our intuition is that a highly mobile user with *FAIR* RSSIs is likely to cause a broken link. Similar to RSSI measurements, we use three levels, $\{$*HIGH, MEDIUM, LOW*$\}$, represent a user's moving speed. Our algorithm use a variable C to track the user speed accumulation. According to the speed level, we increase C with heuristic values (line 9, $\Delta_H > \Delta_M > \Delta_L$). When C exceeds a threshold τ, the path recovery process will be started.

C. Route Cache Management

In MANET routing protocols, every node records the know paths in a route cache to avoid the delay of path establishment. For both path establishment and path recovery protocols, route cache plays an important role. When processing an RREQ message, a node will first check its route cache and if a matching path to the destination is found, the node will reply to the source without further flooding the RREQ. However, the paths in the route cache are not verified until the node decides to adopt them. In a MANET, the link conditions are dynamic and the known paths may not be stable. Using stale paths will cause path recovery once a broken link is detected and yield a worse performance than not using the route cache.

In our solution, we address this issue by removing invalid paths in the cache based on the overheard packets. Two types of packets will trigger a cleansing of the route cache. First, if a node receives a broadcast RERR packet over the long-range radio indicating a broken or about-to-break link, it will search its route cache and eliminate all the paths that contain the link specified in the RERR. Second, every node will listen to the active data transmissions over WiFi or Bluetooth from the neighboring nodes even if the packets are not designated to it. By sniffing these packets (e.g., from node j to node k), node i can measure the RSSI and estimate the quality of the link $j \rightarrow i$. If the RSSI is in the category *POOR*, node i will remove all the paths in its cache that contain the link $j \rightarrow i$.

V. SYSTEM IMPLEMENTATION

In this section, we introduce our implementation of LAAR with off-the-shelf devices. In our prototype, we attach a TinyNode [22], which includes a long-range radio transceiver, Xemics XE1205 [7], to an Android smartphone. XE1205 operates on 915Mhz and feature low cost, low power consumption, and a communication range of 1.6 miles. We have integrated the long-range radio into assorted phones including HTC Magic phone, Nexus One phone, and Nexus 4 phone. We use PL2303 [23] USB-to-Serial bridge controller to connect TinyNode and smartphone (through either ExtUSB or MicroUSB port).

Software support includes programs on both smartphones and the external devices. 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 [24] 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. In our solution, TUN is used for routing, while TAP is used for creating a network bridge. Then we have developed 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. 6 shows two prototype smartphones equipped with TinyNode conducting a ping test with dual radio model.

Fig. 6: Demonstration of the 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 the experiments on basis of a small scale network and NS2 [25] simulation on basis of a large scale network. Our major performance metrics are overhead, number of messages transferred, and network throughput.

We compare LAAR with DSR [10], DSR-R0, and AODV-ERS [26]. DSR-R0 is the default implementation of DSR in NS2 and improves DSR with a *ring-zero search* scheme in the path establishment. Ring-zero search aims to reduce the overhead by firstly sending an RREQ with TTL=0. If the sender and the 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.

Workloads: We consider brochure dissemination application for our evaluation. We collect a set of real brochure files for our tests considering the following cases where a MANET could help disseminate the files. *(1) Advertisements in a mall*: The stores in a mall may want to attract nearby customers by delivering their advertisements or coupons. *(2) Subway map and schedule*: Wireless signals are often poor in subway stations or tunnels. With an effective MANET, the subway administrator can simply deploy a standalone WiFi device to deliver map or schedule files to the commuters without any infrastructure support. *(3) Crowded events*: In an event with a large number of attendees, the infrastructure-based network may have scalability issue because of the limited capacity. [1] With a MANET setting, the attendees can easily check the schedule of shows and other information without connecting to the Internet. Our evaluation uses the sample workload in the following Table I.

TABLE I: Brochure Dissemination Workload

Case	Content	Format	Size
1	Homedepot 20% off coupon	PDF	213KB
2	MTA(New York) Map	PNG	344KB
3	Target Black Friday 2014	HTML	915KB
4	MBTA(Boston) Schedule	PDF	1.2MB
5	AT&T Cyber Monday Sale 2014	PDF	2.1MB
6	NYC Thanksgiving Parade	PDF	2.8MB
7	Nordstrom Anniversary Sale 2014	PDF	4.2MB
8	Mall of America Direction	SWF	5.2MB

A. Experimental Results

First, we build a small scale Ad-Hoc network consisting of 6 Android smartphones equipped with the long-range radio. The phone-to-phone Ad-Hoc mode is supported with WiFi and WiFi-Direct. In our experiments, the smartphones are placed at the fixed positions, i.e., the mobility is not considered. We mainly evaluate the data throughput and the performance of the path establishment protocol. The hop distance between the source and destination ranges from 1 to 5.

Fig. 7 plots the experimental results with the workload in Table I. The bars show the time consumption of the transmission in each case versus the length of the path. Apparently, the overhead grows along with file size and path length. For example, disseminating 2.1MB (case 5) and 4.2MB (case 7) files takes 4.178s and 6.424s respectively for a 2-hop path. The overheads are increased to 25.575s and 30.134s for a 3-hop path. We observe that a MANET is effective for delivering up to a few Megabytes of data to nearby nodes. For a large file over a long path, e.g., case 8 (5.2MB) with a 5-hop path, the overhead may not be acceptable for users. In practice, considering the dense user population and possible user content sharing, we expect a short path length for any communication session. In addition to the overall performance, we also evaluate the breakdown overhead and try to answer the following questions.

Can we use the long-range radio for data delivery? The protocol design could be much simplified if the long-range radio can carry out the data transmission. We have conducted the same experiments with direct transmission between two TinyNode devices. The results are shown in Fig. 8a. Compared to Fig. 7 , the time consumption with the long-range direct link is much higher. Fig. 8b further compares the throughput of direct long-range radio link with hop-by-hop transmission along a 5-hop path. In this experiment, we use "iperf" tool to record the throughput every 20 seconds. We observe that hop-by-hop delivery yields a much higher throughput (with a high variance) even over a long path. Overall, we conclude the long-range radio works well for small management packets, but is not suitable for bulk data transmission.

B. Simulation

In addition, we conduct simulation with NS2 to evaluate LAAR in a large scale network.

1) Simulation Settings: In the simulation, we consider the brochure dissemination application in a mall. We run the simulation in following two settings. **(1) Single store:** In this setting, there is only one store trying to send out brochures to the nearby shoppers. We assume that the store periodically broadcasts a short message including a link to the brochure

---

[1] For example, it was reported that more than 3 millions people attended the 86th Annual Macy's Thanksgiving Day Parade and the explosive users' demand within central park west area caused serious congestions in mobile networks.

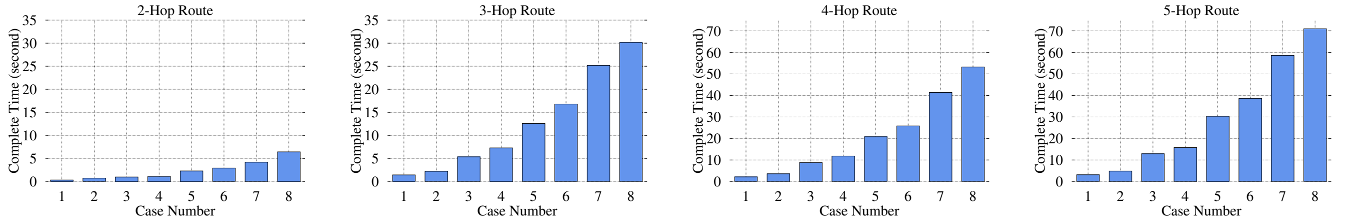
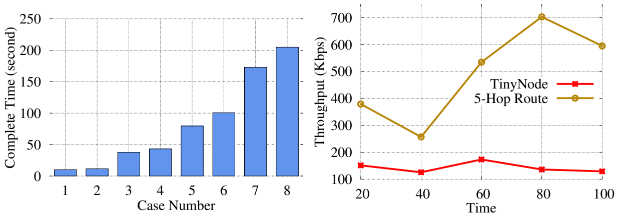


Fig. 7: The experimental performance of overhead with different path lengths



(a) Direct link

(b) Direct link v.s. 5-hop path

Fig. 8: TinyNode data delivery performance

file over the long-range radio. The users can use the link to fetch the brochure. We assume that N users receive the short message and $\alpha \in [0, 1]$ portion of them will be interested in it, i.e., $\alpha \times N$ users will download the brochure. **(2) Multiple stores:** In this setting, there are multiple senders in the mall. Similar to the previous setting, the senders first use periodical short messages over the long-range radio to notify the users.

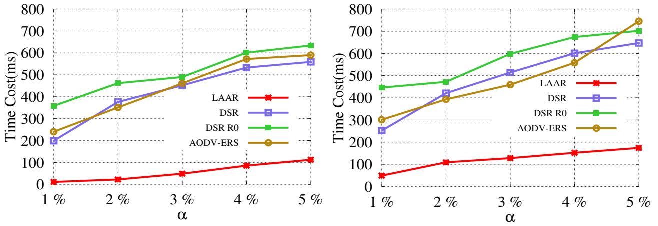
The parameters in NS2 are set as follows. First, we adopt two-ray ground reflection model and constant speed propagation delay model for wireless signal prorogation. In addition, each node in our LAAR protocol is set with two radios. We modify the NS2 to support two wireless interfaces. The frequency of the long-range radio is set to be 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 50m and 100m respectively. For the results shown in this paper, β is set to 0.9 for the RSSI-guided flooding.

The users in the simulations follow a manhattan grid mobility model [27] with a maximum moving speed of 2m/s. At an intersection, the probability of going straight is 0.5 and taking a left or right is 0.25 each. We generate mobility traces with different numbers of users, and in each trace, users randomly select the initial positions inside a store or on a corridor. For all the tested protocols, we set RREQ's default TTL to 5 if applicable. For speed level with user mobility, the three discrete values in Algorithm 3 are defined LOW (<0.5m/s), MEDIUM ([0.5,1.5)m/s), and HIGH (>1.5m/s). In addition, $\Delta_H = 2$, $\Delta_M = 1$, $\delta_L = 0$, and $\tau = 3$.

2) *Single Store:* In this setting, we choose Nordstrom (case 7) as our sender and conduct the simulations with different values of the parameters N and α. For each particular setting, we randomly generate 100 mobility traces for tests, and present the average values in the following figures.

Path establishment: First, we set N to 200 and 300, and change the value of α to control the total concurrent sessions in the network. The overhead performance of initial path establishment is shown in Fig. 9.

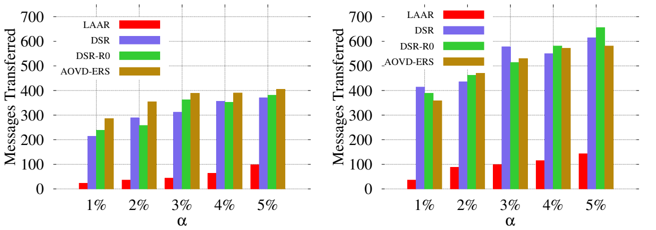
In our setting, the average number of neighbors is 23.9 and 34.3 for $N = 200$ and $N = 300$ respectively. The dense topology can lead to a serious congestion in the existing routing protocols, for example, as shown in Fig. 9b, to construct 9 $(3\% \times 300)$ concurrent sessions, DSR, DSR-R0 and AODV-ERS uses 514ms, 598ms and 459ms, respectively. However, in LAAR, the overhead of path establishment remains low, because our design reduces the number of messages transferred mitigating the effect of congestion.



(a) $N = 200$

(b) $N = 300$

Fig. 9: Average overhead of path establishment (single store)



(a) $N = 200$

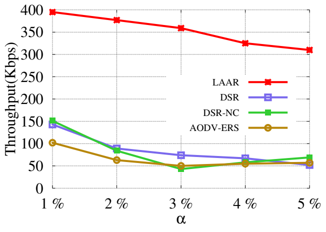
(b) $N = 300$

Fig. 10: Average number of messages transferred to establish path in single store: varying α with 200, 300 users

Fig. 10 illustrates the number of messages (RREQs) transferred in the entire network. The bars indicate a similar trend in all the protocols. Our solution LAAR significantly outperforms the other three protocols.

Overall throughput: We use throughput as an overall performance metric taking full mobility trace and link breaks into consideration. Since DSR and DSR-R0 use the same path recovery protocol, we do not include DSR-R0 in this test. Instead, to better study the impact of stale routes in the cache, we evaluate a DSR protocol that does not use route cache.

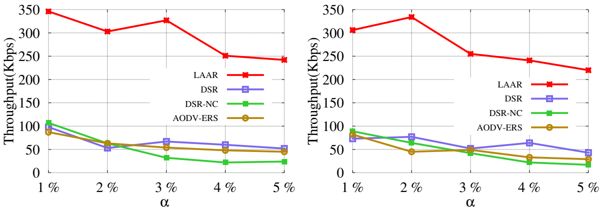
The results are compared in Fig. 11. LAAR maintains a high throughput with different α. For example, with $N = 300$ and $\alpha = 3\%$, the throughputs of DSR, DSR-NC, AODV-ERS and LAAR are, 34.1, 30.7, 50.8, 289.1Kbps. LAAR's throughput is more than five times the throughput of the second best protocol, AODV-ERS. The major reason of the significant improvement is the efficient path recovery protocol in LAAR. Our solution greatly reduces the overhead and congestion during a recovery process, and also improves stability of the selected path and the effectiveness of the route cache.



(a) $N = 200$ (b) $N = 300$

Fig. 11: Average throughput in single store

3) Multiple stores: Finally, we test with three stores, Target (case 3), AT&T (case 5) and Nordstrom (case 7) as our senders. Each store tries to disseminate its brochure listed on Table I. In our configuration, the α for each store's brochure is the same. Thus, the total number of transmissions in the network is $3 \times \alpha \times N$. We collect the throughput from each transmission session and show the average result for each different α value.



(a) $N = 400$ (b) $N = 500$

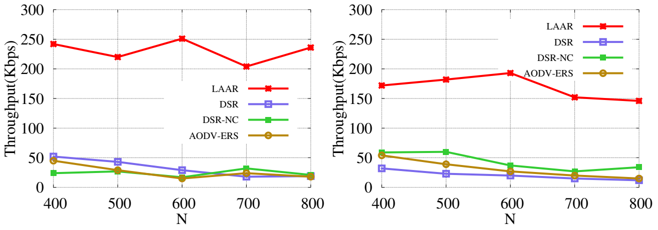
Fig. 12: Average throughput in multiple stores (3 stores)

Overall throughput: Fig. 12 plots the throughput with different values of α when $N = 400, 500$. Obviously, LAAR performs the best among the four tested protocols. For example, in Fig. 12a, the throughputs of LAAR, DSR, DSR-NC, AODV-ERS are 327.4, 67.0, 32.5, 54.4Kbps with $\alpha = 3\%$, respectively. We also find that the throughput of LAAR is not always inversely proportional to the increase of α. For instance, in Fig. 12b, the throughputs are 306.1, 334.4Kbps for $\alpha = 1\%, 2\%$. With more users involved in the transmission, our techniques of proactive path recovery and route cache management wii be more effective helping improve the throughput performance.

Fig. 13 shows the throughput with different number of users (N). The values of α in Fig. 13a and Fig. 13b are set to 5% and 15%, respectively. Again, LAAR outperforms the other three protocols.

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 and recovery which are critical components in the routing protocols. The experimental and simulation results show that LAAR dramatically improves the performance.



(a) $\alpha = 5\%$ (b) $\alpha = 15\%$

Fig. 13: Average throughput in multiple stores (3 stores)

Acknowledgement: This project was supported by National Science Foundation grant CNS-1527336.

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.

[9] Olsr. 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] Y Cheng-Ren et al. Configuring cloud-integrated body sensor networks with evolutionary algorithms. In *Proceedings of the 9th International Conference on Body Area Networks*, BodyNets '14, 2014.

[19] Ying Mao et al. Pasa: Passive broadcast for smartphone ad-hoc networks. In *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*, pages 1–8, Aug 2014.

[20] Yi Cheng-Ren et al. Leveraging evolutionary multiobjective games for configuring cloud-integrated body sensor networks. In *Soft Computing and Intelligent Systems (SCIS), 15th International Symposium on*, pages 630–636, 2014.

[21] Ying Mao et al. Laar: Long-range radio assisted ad-hoc routing in manets. In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pages 350–355, Oct 2014.

[22] Tinynode 584. http://tinynode.com/?q=product/tinynode584/tn-584-868.

[23] Pl2303. http://www.prolific.com.tw/US/newsdetail.aspx?news_id=29.

[24] Tun/tap. http://en.wikipedia.org/wiki/TUN/TAP.

[25] Network simulator 2. http://www.isi.edu/nsnam/ns/.

[26] Woonkang Heo and Minseok Oh. In *FGCN (2)*, pages 128–132.

[27] Manhattan mobility model. http://en.wikipedia.org/wiki/Manhattan_mobility_model.