Mobile Devices Routing Using Wi-Fi Direct Technology

Ricardo Pagoto Marinho, Urbano Botrel Menegato,
Ricardo Augusto Rabelo de Oliveira

Laboratório Imobilis
Universidade Federal de Ouro Preto
Ouro Preto, MG - Brazil
Email: `{ricardopmarinho,urbanobm,rrabelo}@gmail.com`

***Abstract*—Information exchange between mobile devices grows every day. The communication relies on a network with an access point (Wi-Fi, cellular, etc.) using ad hoc communication because Wi-Fi Direct would avoid this dependence. Currently, Wi-Fi Direct does not support multi-hop communication or moving devices. This work focuses on expanding the use of Wi-Fi Direct technology, so that information sent from a device can walk on the network (multi-hop). We measured the number of exchanged messages by devices using four routing protocols: i) flooding, ii) Ad hoc On-demand Distance Vector (AODV), iii) AODV-Backup Route (AODV-BR), and iv) Location-Aided Routing (LAR). We see that even with some challenges one can route mobile devices over Wi-Fi Direct.**

Keywords*–*Wi-Fi Direct; Ad-hoc Network; Routing.

I. INTRODUCTION

Mobile devices have a highlighted place in society life. Many people have some type of mobile device. Therefore, communication using mobile devices is very typical. Apps that help them communicate are frequently used and are popular. However, these apps rely on some sort of network so that the messages can be exchanged, whether it is Wi-Fi or cellular.

These types of networks are already well known and can guarantee that information is delivered even though it may not be possible to use them. Free and good Wi-Fi networks are not easy to find or are not available [1]. In addition, the device must be on the reaching area of a Wi-Fi modem so it can connect itself to the network. Cellular networks can be found anywhere in a city in which they are available, even though access to these networks is made by means of payment whether right on time or before access.

Ad hoc networks can be a cheaper alternative for the devices to communicate. In this type of network, the devices communicate directly with no dependence on the access point as modems on Wi-Fi networks or antennae on cellular networks. In this context, Wi-Fi Alliance has developed Wi-Fi Direct technology [2]. This technology uses a Wi-Fi interface for them to communicate in an ad hoc fashion [3].

This is a new technology and few devices have it. Android devices have this technology. Because Android 4.0 devices are equipped with Wi-Fi Direct technology, we can exploit the technology to communicate between devices. Currently, the communication only happens between two devices that are within the reaching area of both with no hops on communication. Moreover, the devices must be motionless so that communication can be performed efficiently.

Botrel Menegato *et al.* [4] use Android's service API to publish information such as speed, location and battery power. This information can be shared between devices. Based on Android implementation and through experiments, we observed that there is a 24-character limit for the service name being published. We think that 24 characters in many cases are not sufficient to give a relevant name to a service. Very often, 24 characters are not sufficient so that all information on a route message can be sent. Thus, for routing algorithm experiments in some cases, information must be split into more than one publishing.

In this paper, we intend to expand this technology's use on the Android. By doing so, the mobile devices that use Android can communicate in a multi-hop and mobile network way (with devices joining and leaving). To do so, we used four ad hoc routing protocols: i) flooding, ii) AODV [5], iii) AODV-BR [6], and iv) LAR [7]. The flooding protocol is the simplest. With it, the devices only send messages to its neighbors, with any type of control. These messages allow neighbors to know that the devices are on the network. Neighbors, on their turn, replicate with any type of control the information so that every device on the network is aware of other devices. On AODV, AODV-BR and LAR algorithms, devices that want to communicate must initiate a routing discovery stage before sending any message. On this stage, the device that wants to communicate starts a route request with its neighbors. They replicate the message until it arrives on the destination device. The destination then creates the reply message. This message has the route that the request message went through to arrive at the destination. When the reply message arrives on the source device, the route is created. LAR is different from AODV because LAR uses the location and moving information of the destination device to forward requests. AODV-BR is a modification of AODV. The devices maintain a record of the reply messages from other devices so they can produce backup routes for link failure cases.

The experiments were made to measure the number of exchanged messages by every protocol using Wi-Fi Direct. During the experiments, the devices were always in the range of each other. Experiments were performed by varying the number of devices from two to seven. They were performed to determine if the technology can scale and understand its behavior as we added more devices on the network.

The contribution of this paper is to show that, even with technology limitations, we can do ad hoc routing of Android devices using Wi-Fi Direct.

This paper is organized as follows: in Section II, we show several works in literature that also perform experiments on ad hoc network protocols to understand the behavior of the technologies and scenarios used. Section III shows how the protocols are proposed in the literature. In Section IV, we explain how we implemented the proposed protocols on the technology. Section V explains the configurations of the experiments. Section VI shows the results we obtained from the experiments on the technology and Section VII concludes the work.

II. RELATED WORK

Barolli *et al.* [8] propose experiments on Mobile Ad hoc Networks (MANET) using the Optimized Link State Routing (OLSR) protocol. Using eight different scenarios for the experiments, they collect information regarding throughput, Round-trip Time (RTT) and package loss. To do so, they spent 150 seconds on each experiment in a closed environment with all devices reaching everyone. The experiments were performed using OLSR over data flow from TCP and UDP to count the metrics.

Sharma *et al.* [9] use Content Centric Networks (CCN) for communication devices on a MANET to increase the message delivery efficiency on this network. The CCN paradigm only takes into account what the information brings from where it comes. Similar to Barolli *et al.* [8], they also use OLSR to perform the experiments. The proposed algorithm relies on a Multipoint Relay (MPR). The work uses probabilities for a node that is selected based on whether it is an MPR. The experiments were performed on Android devices forming different network topologies. Information regarding the package loss rate, delivery time, network traffic and overhead was collected.

Ikeda *et al.* [10] propose experiments to evaluate throughput and package loss rate on MANETs using OLSR and B.A.T.M.A.N. protocols. The devices were used on two scenarios. One scenario has every device stationary, and other scenario has one of the devices moving. Moreover, the devices were on different floors of the building.

Won-Suk Kim and Sang-Hwa Chung [11] proposed a modification on AODV. They used multi-interface multi-channel (MIMC) wireless mesh network issues (WMN) and adapted the protocol for these situations. The new protocol was named Optimized MMIC ADOV (OM-AODV).

Oki *et al.*[12] verified how much battery power AODV and OLSR consume. They used 14 devices to perform the experiments. The goal was to verify which protocol was better for different situations on solar powered devices. The experiments showed the efficiency of these two protocols with different transmission power and information size.

Liu Yujun and Han Lincheng [13] use a modification over AODV-BR to reduce traffic load when a link failure is detected. They modify RREP messages to make an Extended Hello Message so only neighbors obtain these messages. When a link failure occurs, nodes search over their neighbors so they can find another path to the destination. This modification allows nodes to use fewer control messages to adapt to topology changes.

All of these works focus on testing and experimenting with different types of ad hoc network protocols to determine how the scenarios and technologies that use them behave when using the protocols. This is similar to the work we propose. However, our work uses a technology that to our knowledge, have never been experimented with before, namely, Wi-Fi Direct. In addition, these works differ from ours because we propose a scenario where the communication is all Peer-to-Peer (P2P), while nearly every work presented relies on a certain type of group communication.

III. ALGORITHMS

In this section, we show how the four selected protocols are proposed. The protocols to test the technology are flooding, Ad hoc On-demand Distance Vector (AODV), AODV Backup Route (AODV-BR) and Location Aided Routing (LAR).

These protocols were selected because they are basic protocols for ad hoc network routing. Flooding is the simplest one, AODV only uses the basic route discover approach, and LAR is slightly more complex because it has positioning and movement information. AODV-BR was implemented to test our hypotheses to determine whether a modification on the protocols can make the technology work better. As discussed in Section II, most works used the OLSR protocol, and we did not implement it because these four protocols are sufficient to support our claims.

A. Flooding

For the flooding protocol, the nodes that participate in the network only reply to messages sent by its neighbors without any control on them. Thus, two non-neighbor nodes that have a common neighbor can identify themselves as belonging to the network.

However, with this algorithm, we have the guarantee that the message will go through the best path between two nodes because every node on the network will receive it.

B. AODV

The AODV protocol is a reactive protocol. On this type of protocol, a node will know if there is a route to a destination after a route discovery phase.

This phase begins when a node s wishes to communicate with another node d on the network. If node s does not have a route to d, the route discovery phase starts. First, node s sends its neighbors a Route Request (RREQ) message through a broadcast. The message has the id from s and d and the route on which the message passes through (with the source id - s). The message is replicated by the intermediate nodes through a broadcast. These nodes add their ids to the route field to form a reverse route used by the reply message to arrive on s. The reply message is created in two cases: i) the message arrives on the node d, or ii) the message arrives on a intermediate node that has a route to d.

In these two cases, a Route Reply (RREP) message is created. This message has the id form s and d and the route on which the RREQ message went through to arrive on the node and where the RREP message must return to arrive on s. When an intermediate node has a route to d, it adds its id and the route that it has to d on the message.

Figure 1 shows the first process. First, node F wishes to create a route to node D. Node F sends an RREQ message to its neighbors, in this case, node X with its id - F. Node X adds its id to the route and sends the message to D. When node D receives the message, it creates an RREP message and sends it through the reverse route. Then, it sends the message to X and then to F, creating the route.

On the second process, X already has a route to D. When it receives an RREQ message from D, it just appends its route on the message route field and creates an RREP message to F.

If some node leaves the network whether by a connection problem or it really leaves the network, a Route Error (RRER) message is created. The message is sent by the node that identified the problem to notify the nodes that participate in the network so that the problematic node is not available. Thus, as soon as a node receives this message, it removes every route that has the broken node from the routing table.

C. AODV-BR

This protocol is a modification of classic AODV. With this modification, the nodes produce alternate routes to a destination. When a node receives the RREP message, it stores the route information from the message, even if it already has a route to the destination. With this, other routes to the destination are formed when a link failure occurs.

In this paper, we use this idea to obtain route information from RREP messages. However, unlike AODV-BR, we use the information for a node to decide if it will send its RREP message.

When the node receives some RREP message, it stores information regarding the node that the message is for - who has begun the route discovery, and the hop numbers for that route.

With this information in hand, when the node wishes to send an RREP message, it first checks if there is already a route for that destination.

When a route is not known, the node sends its message. However, if a route is already known to the destination, the node checks if its route is better - in number of hops - than the ones it has stored - received from other nodes. If so, it sends the message. If not, it does not send it.

D. LAR

The LAR protocol, similar to AODV, is a reactive protocol. It relies on RREQ and RREP messages to create routes even though it differs from AODV by using devices' geographical positioning information on the network to communicate.

When a device s wishes to communicate with another device d, it first needs information of where, at t_1 instant, d was. This information is geographical positioning information (latitude - X_d - and longitude - Y_d), direction (D_d) and movement speed (V_d). Based on this information, when s wishes to make a route to d on a t_2, instant it calculates a possible area where d can be on t_2. This area is a circle centered on X_d and Y_d with a radius $V_d(t_2 - t_1)$.

Once this area is created, a request is sent to s's neighbors. When one of its neighbors receives this message, it verifies whether it is on the specified area. If it is, it keeps propagating the request; if not, it discards the message. The reply happens the same way on AODV.

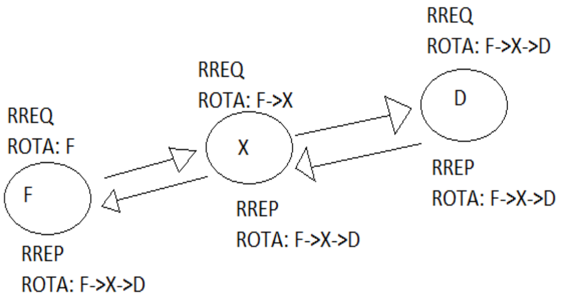


Figure 1. Route creation process.

To make more devices propagate the request, we can increase the possible areas where d might be. This area makes a rectangle, where s and the possible area that d might be are on a diagonal. In this way, the rectangle will have (X_s,Y_s), $(X_d + V_d(t_2 - t_1), Y_s)$, $(X_d + V_d(t_2 - t_1), Y_d + V_d(t_2 - t_1))$ and $(X_s, Y_d + V_d(t_2 - t_1))$ as the edges. These coordinates have where s is and the circle that d might be: $(X_d + V_d(t_2 - t_1)$ and $Y_d + V_d(t_2 - t_1))$.

IV. IMPLEMENTATION

To test the technology on the proposed environments, we used Android, Wi-Fi Direct, and service publishing functions.

The Wi-Fi Direct technology offers functions to recognize nearby devices using it, as search and automatic identify, and elects a Group Owner (GO) to manage the network [4].

A. Framework

Botrel Menegato *et al.* [4] created a service publishing framework to elect cluster heads. So, they wish to make the GO election more reliable for the elected node relevant inside the cluster context.

They used all functions offered by the Android API to Wi-Fi Direct and service publishing, including scanning the network for devices and receive messages sent by neighbors. Thus, the information regarding the cluster head was sent to neighbor nodes so they could decide which device was the best for the job.

Information regarding devices on the network was published for available services they had. These published details are strings with the information that devices wish to send, even though it has size limitations that make it possible to send only a small amount of information (approximately 24 characters) at once. In addition, published services are continuously sent by the API, ending only they when are explicitly requested. Other limitation is that there is a limit on the number of different services being published by one device. Each one can have approximately seven different services. Once this number is reached, it is necessary to end some service publications to start another one.

Information published in [4] was only sent to neighbors and to those that do not send information to their neighbors. This issue opens another use for the framework. We can use it where information must be sent to other nodes beyond neighbors such as in routing.

B. Protocols

To implement the protocols, we made every message they must send a service that is published by the device. This means that when a device wishes to create a route to another device, the RREQ message is a service with all information on it, such as the type of message - RREQ, the source and destination device ids and the route. When an intermediate device receives and forwards a message, such as a device forwarding an RREQ message, it also publishes the message it is making available as a service, even though the process was not initiated by it. So, when this occurs, the device has its own services, as requests and replies initiated by it, and services from another device published.

When choosing a device to initiate a request, the devices have a table that stores ids from other devices on the network. This table is filled when a device enters the network and publishes a service (message), informing the other devices that it is on the network. With this table containing at least one id on it, the device starts the route request process. When more than one id is on the table, it chooses at random the one to which the request will be made.

As we said before, the service publishing API available for Android has limitations. When a routing information does not fit in only one message (has more than 24 characters), it is split into additional messages so that all of the information can be sent. Putting this together with the services that continue being published by the API, the network has more load.

Thus, to build the RREQ and RREP messages, one must use numbers to show the number of messages that will be necessary to send the information and to show where the received message is, together with the usual routing information as source, destination and route. For example, if a message must be split into two, the first one has the numbers two (total amount of messages) and one (showing that this is the first one). The second has the numbers two and two (total amount and second message).

When a device receives a message with a number greater than one for the total number of messages to be received, it keeps this information until the next one arrives. As soon as it arrives, the information is stored.

A problem that may occur is the messages arriving out of order. In this case, the second message is stored and completed when the first one arrives.

In addition, because the LAR implementation has more information than AODV, such as geographical positioning and direction information, we had to split it into additional messages so all of the information could be sent, causing more challenges to the received message control.

To use the work from [4], we had to expand its functionalities. So, it was necessary to make modifications when a device receives a request and sends the message, as discussed in Section III.

V. EXPERIMENTS

Experiments were performed to verify Wi-Fi Direct's scalability when we introduce more devices for communication. For this, we used from two to seven tablets: five Samsung Galaxy Tab 2 (three of them as an operational system Android 4.1.1 and two as an Android 4.1.2) one Samsung Galaxy Tab 3 with Android 4.2.2 and one Samsung Galaxy Note with Android 4.3.3.

The experiments were performed with every tablet in the range of all of them so that the scalability could be tested. On the experiments, we counted the number of messages exchanged on each one of the protocols previously cited to test how the technology behaves with them.

Every time an experiment was performed, we switched off the Wi-Fi interface for the messages to stop publishing, avoiding the interference with the next experiment. After this, the application on the tablets was initiated and the message number was counted. Each experiment took between 15 and 20 minutes to be completed.

VI. RESULTS

Here, we show and discuss the measuring results by graphics.

A. Flooding

On the flooding protocol implementation, we only used one type of message. This message tells the neighbors about the existence of a node on the network. As previously discussed, as soon as the neighbors receive this message, they reply to their neighbors, so every node on the network can be aware of the participating nodes on the network. By doing this, the measures were made by considering the number of sent and received messages for a node. Figure 2 shows the means of the sent messages by tablets for the experiments.

By analyzing Figure 2, one can see that as we put more tablets on the network, the sent messages mean keeps growing. This shows that the technology bears the introduction of more devices on the network for sent messages.

We can also see that this message increase is nearly linear. This shows that up to where we made measurements, the technology does not modify its behavior as we introduce new devices. The figure also shows the standard deviation on the measures. They are small and tell us that the variation on the measures was small. Because the variation is small for every measurement, we can state that the technology does not change its behavior as we introduce more devices on the network. This confirms our affirmation that for this type of message, the technology bears the introduction of more devices on the network and is stronger.

By analyzing Figure 3, one can see that for up to four tablets on the network, the mean rises linearly. This shows that the technology bears, without any problem, received messages from four tablets at the same time. With more tablets, the network starts to deteriorate, *i.e.*, lose its ability to receive messages, and with six and seven tablets, the deterioration was greater.

The network deterioration can be better shown in Figure 4. As previously discussed, the technology sends many messages until the service is ended. This makes the number of received messages great than sent messages.

With this said, we made Figure 4 by using the difference between the number of received and sent messages. Along with everything that was discussed in the previous paragraph, we can wish that for five tablets, more messages are received than sent so that the figure will have a negative number. At this point, the difference between sent and received messages should rise and the numbers should become lower. However, for six or more tablets, for the point where the network deteriorates, this number should be positive.

As we thought, the difference between sent and received messages grew up to five tablets after this because the network deteriorates the difference invert and more messages are sent than received.

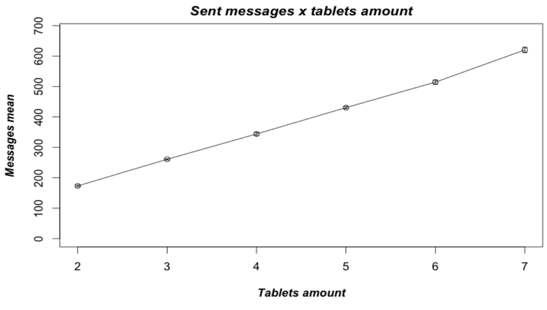
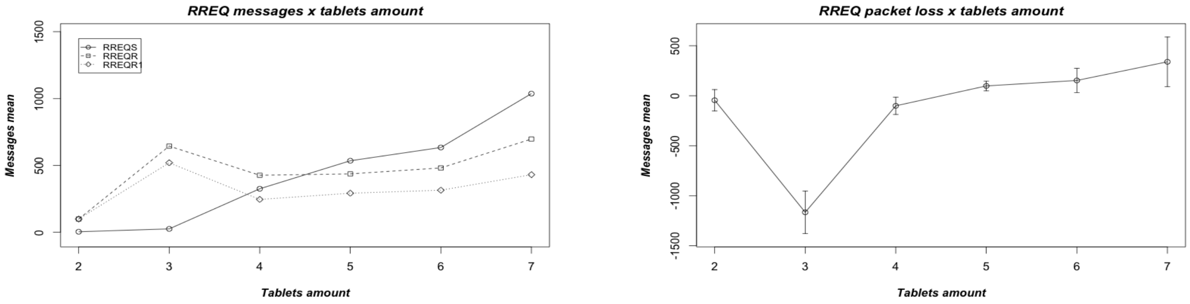
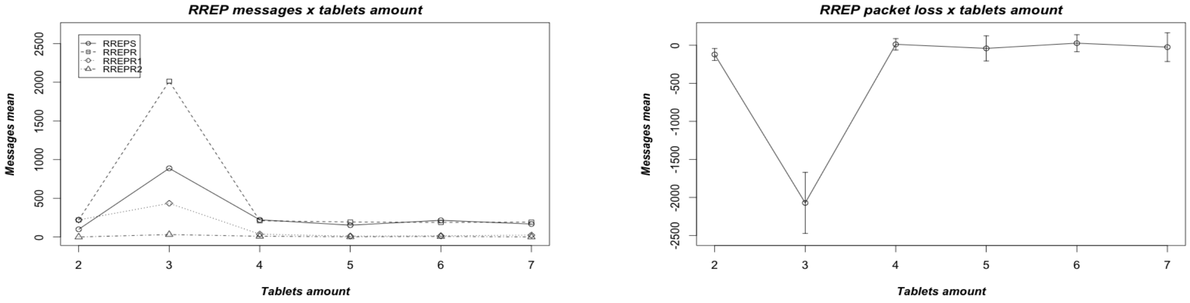


Figure 2. Sent messages x Tablets number on flooding.



(a) Comparison between shared RREQ messages on AODV. (b) RREQ packet loss x Tablets number on AODV.



(c) Comparison between shared RREP messages on AODV. (d) RREP packet loss x Tablets number on AODV.

Figure 5. AODV comparisons.

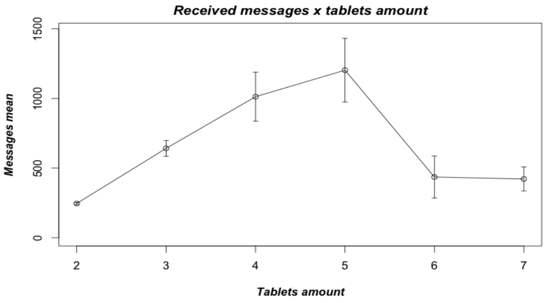


Figure 3. Received messages x Tablets number on flooding.

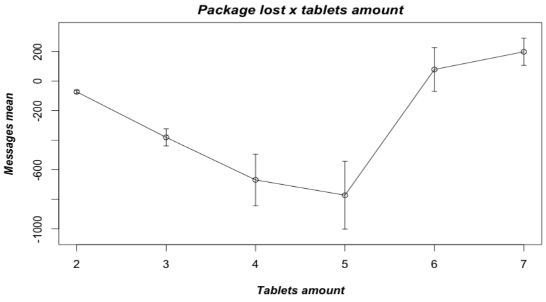


Figure 4. Packet lost on flooding.

B. AODV

AODV has three types of messages, namely, RREQ, RREP, and RERR. RERR messages were not taken into account because every tablet was in the range of all of them, and this type of message is not generated very oftem. RREQ and RREP were split into several categories because all of the messages are not applicable to a device. An example is an RREQ message arriving at a device, while the request was not for it. So, the RREQ message measures were split into sent (RREQS), received (RREQR), and received messages on which the destination was the device. This message was tagged as type 1 received RREQ(RREQR1).

Not every RREP message is relevant for a device. Consider the case where a device receives an RREP message, but the device is not on the reverse route; the message has no meaning for it. So, they were split into measurements for sent (RREPS), received (RREPR), received so that the device made the request (type 1 - RREPR1) and received so that the device did not make the request but belongs to the reverse route (type 2 - RREPR2).

Figure 5a shows a comparison for RREQ messages. For received RREQ, one can see that the network begins to deteriorate after three tablets are on it. Here, the tablets receive nearly all of the sent RREQ messages. However, after this point, all received message measures drop. An interesting fact is that with seven tablets, the network received nearly the same number of messages, when there are three tablets on it. This can show us a technology boundary. To confirm this statement, experiments with more tablets are needed.

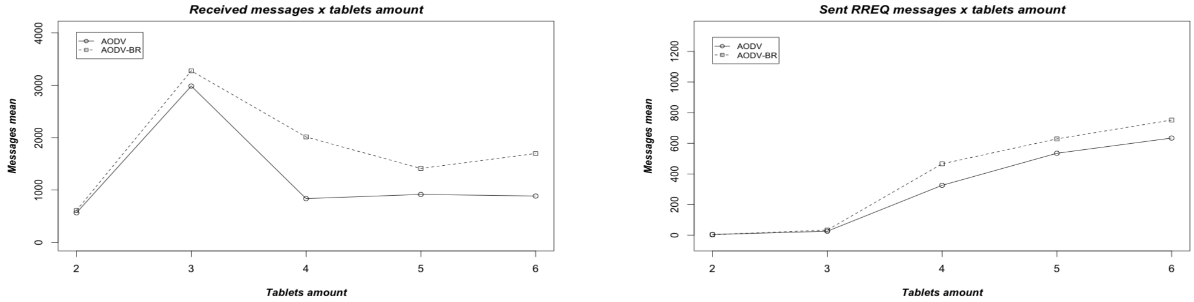
Type 1 received RREQ messages have the same behavior as the received RREQ messages. This occurs because type 1 received RREQ messages are within the received RREQ messages.

Comparing the curves in Figure 5a, when we introduce more than four tablets, the network begins to deteriorate. While the sent messages curve always grows, the curve of the received messages drops when it arrives at four tablets and starts to grow when we have more tablets on the network.

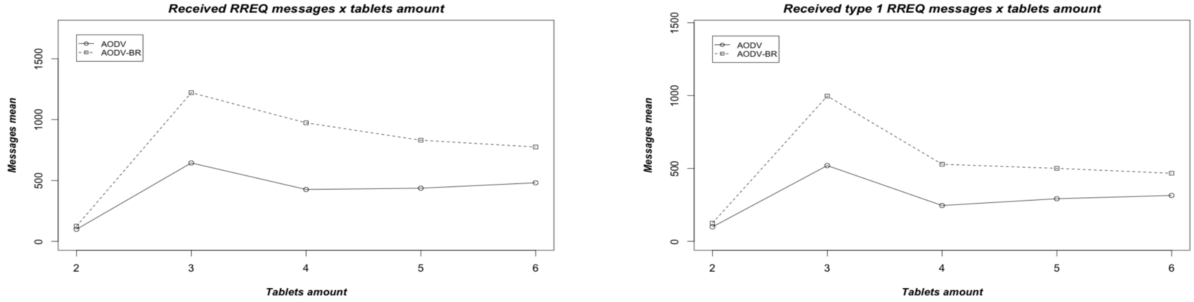
Figure 5b shows the difference between sent and received RREQ messages. We hope that for two and three tablets, the curve grows negatively and for four or more tablets, it begins to grow positively because of network deterioration.

As previously discussed for two and three tablets, the technology receives the most messages and their copies are made by the service API. With five or more tablets, the number of sent messages is greater than the received messages.

Figures 5c and 5d show the same comparisons made with RREQ messages for RREP messages.

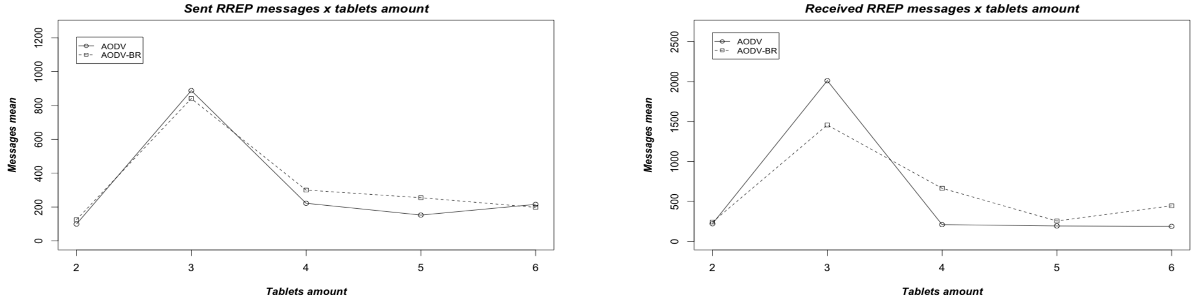


(a) Received messages number x Tablets number. (b) Sent RREQ messages x Tablets number.

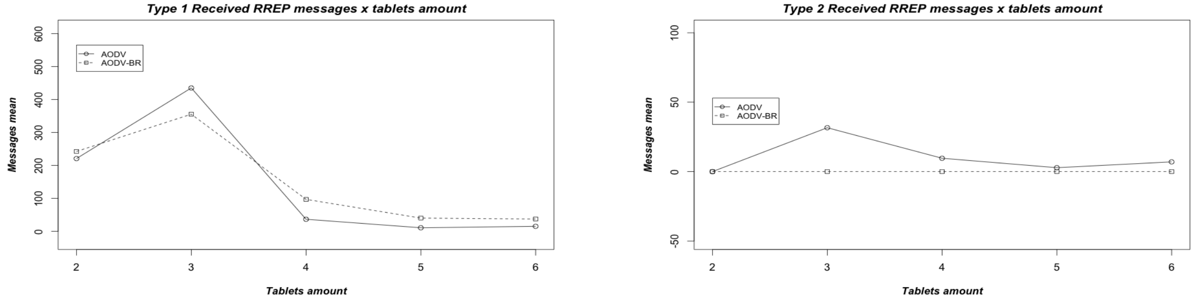


(c) Received RREQ messages x Tablets number. (d) Received type 1 RREQ messages x Tablets Number.

Figure 6. AODV/AODV-BR comparison - 1.



(a) Sent RREP messages x Tablets number. (b) Received RREP messages x Tablets number.



(c) Received type 1 RREP messages x Tablets number. (d) Received type 2 RREP messages x Tablets number.

Figure 7. AODV/AODV-BR comparisons - 2.

By analyzing Figure 5c, one can see that for every type of message, its mean drops when there are four tablets on the network. This differs from flooding and RREQ messages measures. This occurs because RREP messages are created only after an RREQ message arrives. As Figure 5a shows, when there are four tablets, the receiving process of the RREQ messages is affected, causing problems for the RREP messages. Moreover, Figure 5c shows that with four or more tablets, the sent messages are nearly equal to the received messages. We can state that with four tablets the network begins to deteriorate and affects message receiving. Figure 5d shows this comparison in a better way. This figure shows that the network receives the most messages with three tablets and that this is the lower measured value. From this point on, the network begins to deteriorate and the comparison differences are closer. This figure's behavior differs from others because the messages rely on receiving RREQ messages.

C. LAR

For LAR experiments, we realized that Wi-Fi Direct could not receive and send messages for a long period of time (it worked for approximately 1 minute).

Because the API has limitations, *e.g.*, the number of services published, we performed experiments by putting a limit on this number to determine how long Wi-Fi Direct would work. The results are in Table I.

The table shows a possible limit on the Wi-Fi Direct working period when used for routing purposes: 15 minutes. This value was seen for the AOVD experiments, while on flooding, the time was 20 minutes. We can state that the amount of service being published at the same time affects the technology working time.

D. Comparison: AODV/AODV-BR

Here, we will show the results when we compare AODV and AODV-BR. Our goal in making this modification is to decrease the number of messages and keep the network working better for more time, *i.e.*, prevent the network from deteriorating with four tablets or make the deterioration smoother.

Figure 6a shows the difference between received messages on AODV and AODV-BR. It shows that on both protocols, the network has deteriorated when we have four tablets on it. However, with the modification on AODV-BR, one can see that the deterioration was smoother and that with six tablets, the number has grown.

Figures 6b, 6c and 6d show a comparison of sent, received and type 1 received RREQ messages. For sent RREQ message, AODV-BR increased the number of messages. This also happens with the others types of messages, but for received and type 1 received messages, the deterioration was smoother because it occurred with received messages.

Figures 7a, 7b, 7c and 7d show the measuring results with RREP messages. The results were better than the results with RREQ messages. On every figure, one can observe that the number of messages for three tablets was lower on AODV-BR than AODV. We can see that with four tablets, the deterioration was smoother. An interesting point to observe is that on Figure 7d, there were no messages on AODV-BR. With this modification, the device is not concerned about this type of message, when everyone is on everyone else's range.

VII. CONCLUSION AND FUTURE WORK

Analyzing the results and graphics, we can conclude that for small networks (two or three devices), the technology can bear the routing load. With more devices, routing is affected because of the great load on the network. However, with some modifications on the algorithms, such as making the devices aware of other routes in AODV-BR, this issue can be solved. However, even with these problems, it is possible to route devices on an ad hoc network using Wi-Fi Direct technology.

In addition, we found that the technology is affected when we publish different services at the same time by making it stop publishing and receiving information.

A technological contribution is suggested in that the number of characters on the name of the service being published should be greater than 24.

For future work, we will experiment whit the technology in new scenarios, such as for devices moving, joining and leaving the network and for protocols that take other parameters into consideration, including social parameters.

TABLE I. AMOUNT OF SERVICE X WORKING PERIOD ON LAR.

Quantidade de serviços	Tempo (min)
Sem limitação	1
10	5
8	15
6	15

REFERENCES

[1] L. V. Hoang and H. Ogawa, "A platform for building ad hoc social networks based on wi-fi direct," in Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on, Oct 2014, pp. 626–629.

[2] "Wi-fi direct | wi-fi alliance," http://www.wi-fi.org/discover-wi-fi/wi-fi-direct, retrieved: January, 2014.

[3] "Wifi p2p technical specification v1.2," http://pt.scribd.com/doc/215283500/WiFi-P2P-Technical-Specification-v1-2, retrieved: January, 2015.

[4] U. Botrel Menegato, L. Souza Cimino, S. E. Delabrida Silva, F. A. Medeiros Silva, J. Castro Lima, and R. A. R. Oliveira, "Dynamic clustering in wifi direct technology," in Proceedings of the 12th ACM International Symposium on Mobility Management and Wireless Access, ser. MobiWac '14. Montreal, QC, Canada: ACM, 2014, pp. 25–29.

[5] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on. IEEE, Feb 1999, pp. 90–100.

[6] S.-J. Lee and M. Gerla, "Aodv-br: backup routing in ad hoc networks," in Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE, vol. 3. Chicago, IL: IEEE, Sep 2000, pp. 1311–1316.

[7] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," Wirel. Netw., vol. 6, no. 4, Jul. 2000, pp. 307–321.

[8] L. Barolli, M. Ikeda, F. Xhafa, and A. Duresi, "A testbed for manets: Implementation, experiences and learned lessons," Systems Journal, IEEE, vol. 4, no. 2, June 2010, pp. 243–252.

[9] P. Sharma, D. Souza, E. Fiore, J. Gottschalk, and D. Marquis, "A case for manet-aware content centric networking of smartphones," in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, June 2012, pp. 1–6.

[10] M. Ikeda, E. Kulla, M. Hiyama, L. Barolli, and M. Takizawa, "Experimental results of a manet testbed in indoor stairs environment," in Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on, March 2011, pp. 779–786.

[11] W.-S. Kim and S.-H. Chung, "Design of optimized aodv routing protocol for multi-interface multi-channel wireless mesh networks," in Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, March 2013, pp. 325–332.

[12] O. Oki, P. Mudali, M. Mutanga, and M. Adigun, "A testbed evaluation of energy-efficiency of routing protocols in battery-powered wireless mesh networks," in AFRICON, 2013, Sept 2013, pp. 1–7.

[13] L. Yujun and H. Lincheng, "The research on an aodv-brl to increase reliability and reduce routing overhead in manet," in Computer Application and System Modeling (ICCASM), 2010 International Conference on, vol. 12. IEEE, Oct 2010.