Towards the Opportunistic Combination of Mobile Ad-hoc Networks with Infrastructure Access

João A. Silva João Leitão Nuno Preguiça João M. Lourenço Hervé Paulino
NOVA Laboratory for Computer Science and Informatics & Departamento de Informática
Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa, 2829-516 Caparica, Portugal
jaa.silva@campus.fct.unl.pt
{jc.leitao,nuno.preguica,joao.lourenco,herve.paulino}@fct.unl.pt

ABSTRACT

One of the main characteristics of mobile ad-hoc networks (MANETs) is the lack of global, consistent, and up-to-date knowledge of the network topology. Thus, when routing messages, they must be forwarded from one node to the next based solely on each node's current local knowledge of the network. If, somehow, some nodes also have Internet access (even if intermittently), the mix of MANETs with that infrastructure access allows for a wider range of possibilities. In this exploratory work-in-progress paper, we argue for the opportunistic combination of ad-hoc networking with infrastructure access as a way of enabling possible optimizations. The routing protocol can leverage on the fact that some nodes might have infrastructure access and use them to make messages "jump" through the network whenever it pays off. Thus, we address the interaction between the ad-hoc routing and the infrastructure access by devising a decision algorithm that determines when it is better for a message to be routed through the network using ad-hoc techniques, and when it is better to route them through a tunnel where the endpoints are nodes with access to the infrastructure, enabling long "jumps" over the network.

CCS Concepts

•Networks → Routing protocols; Mobile ad hoc networks;

Keywords

Mobile Ad-hoc Networks; Routing Protocol; Infrastructure Access; Mobile Edge Clouds

1. INTRODUCTION

Mobile ad-hoc networks (MANETs) are formed dynamically by mobile nodes that are connected wirelessly without resorting to a pre-existing network infrastructure (i.e., no base stations) [2]. Thus, interaction among nodes is achieved through the wireless broadcast medium without any central coordination entity (in a peer-to-peer fashion). Nodes can move freely, thus the network topology may change rapidly and unpredictably. Furthermore, the lack of a central coordination entity makes routing in MANETs a challenging task. Nodes lack a global, consistent, and up-to-date knowledge of the network topology, being required to make routing decisions based only on local (and potentially partially incorrect) knowledge.

However, given the increasingly ubiquitous Internet access through other technologies that co-exist alongside ad-hoc networks (e.g., Wi-Fi and 3G/4G cellular networks), some of these nodes might also have simultaneous access to a network supported by infrastructure. This uncovers several opportunities when devising routing strategies, allowing the opportunistic combination of ad-hoc networking with infrastructure access. Therefore, when routing messages, two approaches can be employed: one entirely in the ad-hoc network, and a second one that makes use of the access to the infrastructure. So, although the ad-hoc network must be entirely self-supporting (e.g., for emergency situations), it can leverage the infrastructure (when present) during normal operation.

Mobile devices, such as smartphones or tablets, are natural examples of nodes that may have simultaneous access to both the Internet and to an ad-hoc network. The proliferation of this kind of devices, along with the increasing growth of their capabilities, has spawned research on the adaptation of MANET techniques for the mobile devices world (e.g., Serval [17], SPAN [18], and others [19]). Additionally, recent work [16, 20] is exploring how the connectivity capabilities offered by these devices (e.g., Wi-Fi Direct and Bluetooth) may facilitate the use of MANET-like communication among them, with the ultimate goal of creating mobile edge clouds [6].

In this paper, we propose a way of capitalizing on this double access—to the ad-hoc network and the infrastructure—as a way to potentially improve communication and energy efficiency in mobile edge clouds. During the process of routing messages, the routing protocol has to decide which alternative to use: only through the ad-hoc network; or using the infrastructure access, making messages "jump" through the (ad-hoc) network. Thus, we propose a decision algorithm that determines the best path for each message. Figure 1 illustrates the basic idea of the proposed approach, showing two different possible routing paths. The dashed arrows represent a possible path followed entirely in the ad-hoc network, while the full (and dotted) arrows represent the path followed when using the infrastructure access. In the second

MECC '16, December 12-16 2016, Trento, Italy

ISBN 978-1-4503-4668-9/16/12...$15.00

DOI: http://dx.doi.org/10.1145/3017116.3022873

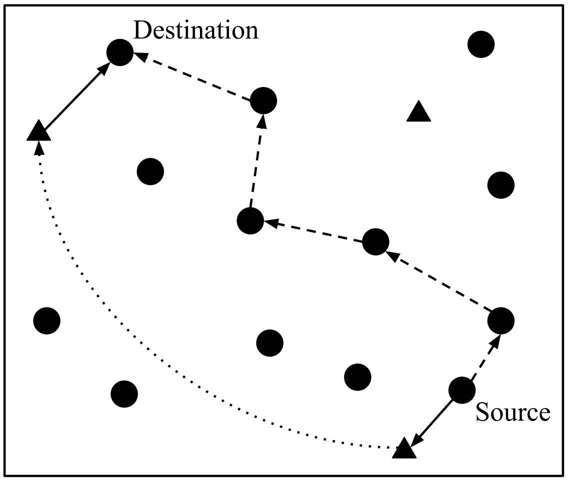

Figure 1: Example of a message's possible routing paths. Circles are regular ad-hoc nodes and triangles are nodes that also have infrastructure access.

case, the routing path is shortened by a considerable amount of hops by using two nodes that can communicate directly between them (using the infrastructure access). This can entail a possible decrease in latency, by avoiding the long ad-hoc hop-by-hop routing. Even otherwise, benefits may arise from the reduction of the overall aggregate energy costs of routing the message through all the intermediate nodes in the ad-hoc network.

The remainder of this paper is organized as follows. Section 2 presents our proposed approach; in Section 3 we elaborate on how we will evaluate it; Section 4 discusses related work; and Section 5 presents our conclusions and prospective future work.

2. COMBINING AD-HOC NETWORKS WITH INFRASTRUCTURE ACCESS

When routing messages following our proposed approach, it is clear that messages between opposite sides of the network will most probably be routed via the infrastructure rather than through the ad-hoc network which might incur in an excessive number of hops and energy spent along the routing path. Likewise, messages addressed to one or two hops away will be routed directly within the ad-hoc network. The cases in between these two extremes however, are not so easily decided, since one has to take into account a significant number of trade-offs, while making the decision to rely only on ad-hoc routing or to leverage opportunities to route messages through the infrastructure. In this section we address the problem that comes from mixing ad-hoc networks with access to the infrastructure—which path should the routing protocol use when routing messages through the network. This includes exploring adequate trade-offs regarding communication latency and energy drainage on devices. Particularly, in some scenarios it might be preferable to route messages through the infrastructure (even at the cost of increased latency) to avoid the prohibitive energy costs of routing them through many hops in the ad-hoc network.

Focusing on the mobile edge cloud context, the type of applications we are aiming for is multimedia file sharing on social events, such as sports events, business meetings, or family reunions. In such scenarios, the mobile devices forming the edge cloud are confined to a geographic space known beforehand, e.g., a football stadium. Mobility is however unrestricted, as devices may move freely within the venue, and even leave its premises (and with that abandon the edge cloud).

These characteristics are favorable to the use of geographic routing protocols [9], which build up from the notion of *node geographic location*, allowing us to have a more concrete and realistic metric for distance to the message destination (that the number of hops in the network), since location bears a close connection to topology in wireless networks.

2.1 Geographic Routing Protocols

Routing protocols for ad-hoc networks are divided into two main categories: proactive (table-driven) and reactive (on-demand) protocols. Proactive protocols (e.g., OLSR [10]) continuously try to maintain topology information up-to-date in every node, by periodically disseminating that information throughout the network. In theory, the whole network should be known by all nodes, which results in a constant overhead of control traffic, but no initial delay in communication. On the other hand, reactive protocols (e.g., AODV [14]) only construct routes to destinations as they are required. Before initiating communication, a node first has to establish a route to the desired destination, which results in some initial delay in communication. Hybrid protocols that try to combine the advantages of both proactive and reactive protocols have also been proposed (e.g., ZRP [8]).

Another possible approach is geographic (or position-based) routing, whereby nodes send messages to the geographic location of the destination instead of using the network address. This requires each node to be able to determine its own location and also that it becomes aware of the location of any destination node when sending a message. With this information a message can be routed to the destination without knowledge of the network topology or a prior route discovery. Routing can be achieved through different strategies. Greedy forwarding is one of the simplest strategies to achieve this. In this case messages are routed to a node that minimizes (at each hop) the distance to the final destination (using only local information).

As described earlier, geographic routing requires nodes to know the location of the message destination. To achieve this, geographic routing protocols require a geographic location service [13]. Some approaches proactively flood node location updates throughout the network, while others reactively flood location queries throughout the network when a node wants to find the location of a destination. Some, arguably more complex approaches use nodes as location servers that maintain, and spread through the network, location information on behalf of some other nodes.

2.1.1 Cell Hash Routing

In our proposal we aim at integrating our algorithm that decides between the use of ad-hoc network and infrastructure communication into a geographic routing protocol—namely, cell hash routing (CHR) [1].

CHR is a cluster-based distributed hash table (DHT), where space is divided into equally sized squares or cells and each cell acts as a virtual node that represents all the real nodes that are inside it. Since nodes need to map any point in space to its corresponding cell, the cell size and a unique origin of space must be agreed beforehand between every node. The cell size should maximize the probability that a node in a cell can listen to all the other nodes in its own cell and some nodes in each of the eight neighboring cells, while at the same time avoiding to be too small such that the cluster-based approach yields no gains.

In CHR, nodes do not need to have a precise notion of location. Instead, it suffices for them to know their cell and be able to reach at least one neighbor in each of the populated adjacent cells. Routing is done at cell-level using a variation of the greedy perimeter stateless routing (GPSR) protocol [11].

Since CHR works as a DHT, it stores $\langle key, value\rangle$ pairs in its cells. The cell responsible for a pair $\langle k, v\rangle$ depends deterministically on the result of applying a hash function to key k. Due to this, and because nodes are clustered into cells, nodes are not individually addressable (because routing works at cell-level). But this can be easily surpassed by using the DHT as its own geographic location service (mapping node identifiers to their locations).

2.2 System Model

We consider a classical asynchronous model comprised by $\Pi = \{n_1, \ldots, n_k\}$ nodes, with no mobility restrictions (assuming the maximum velocity a human can achieve), of which $\Gamma = \{n_1, \ldots, n_j\}$ nodes (where $\Gamma \in \Pi$) have infrastructure access and are called jumper nodes (JNs). Regular nodes can become JNs at any time and vice-versa. Infrastructure access means unrestricted access to the Internet by any kind of network infrastructure, e.g., cellular infrastructure like 3G/4G, or typical Wi-Fi access points (APs).

We assume that nodes communicate by exchanging messages through wireless networks, and have no access to any form of shared memory. The only exception is an external component—a cloud rendezvous point (CRP)—which is known a priori by every node and that runs in the Internet. This component is used by the JNs to store a small amount of information about themselves. We also consider the classical crash-stop failure model, where nodes can fail by crashing but do not behave maliciously. Figure 2 depicts a system overview with the cells and the CRP, and messages being routed both through the ad-hoc network and through the infrastructure.

2.3 Jumper Nodes

Contrary to regular nodes that only have access to the ad-hoc network, JNs have simultaneous access to both the ad-hoc network and the infrastructure. Thus, they can use the latter to communicate directly with other JNs and create tunnels through the ad-hoc network.

When a node detects that it has infrastructure access, it registers itself in the CRP by sending a message of the form

$$\textsc{JnUpdate}[aId, ip, cId, ts]$$

where aId is its ad-hoc network identifier, ip its public IP address in the Internet, cId its cell identifier, and ts an update timestamp. In order for the CRP to have up-to-date information about the JNs and to know if they are still alive, JNs periodically refresh their information and send JnUpdate messages to the CRP.

As part of the regular routing protocol, every node broadcasts periodic beacons with its cell identifier. So, when infrastructure access is detected, JNs also add a flag to their beacons indicating their special status to their one-hop neighbors.

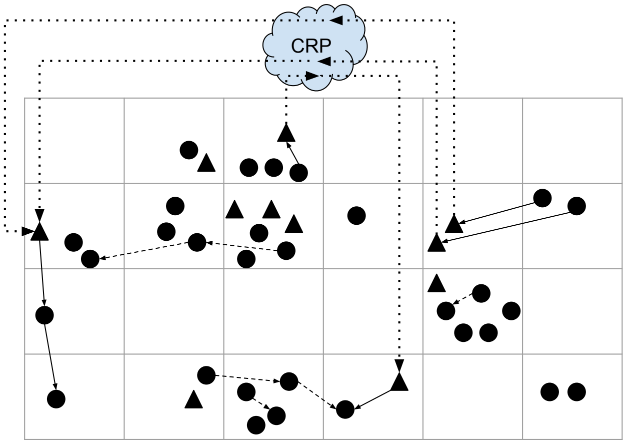

Figure 2: System overview. Circles are regular ad-hoc nodes and triangles are nodes that also have infrastructure access.

A consequence of their special status is that most probably they will be more contacted (on average) than regular nodes. Since this can become unfair in terms of battery drain, JNs must have a way of becoming regular nodes (and conserve battery). Thus, when their battery level drops below a configurable value MIN_BAT, they change their status into regular nodes (even if they still have infrastructure access) by sending a

$$\textsc{JnDelete}[aId]$$

message to the CRP and return to send regular beacons (without the JN flag). Naturally, JnDelete messages are also used when a JN detects it no longer has infrastructure access, and unregisters from the CRP.

2.4 Cloud Rendezvous Point

The cloud rendezvous point (CRP) is a new element in the network that resides in a cloud in the Internet, or a cloudlet, and that is known beforehand by every node in the network. To avoid being a single point of failure and to avoid overloading it, classical replication and load-balancing techniques out of the scope of this paper can be employed. It encapsulates the best path decision logic (see Section 2.5.2) and works essentially as a database for the existent JNs in the network (the central coordination entity absent in regular MANETs). It does not measure latencies, distances, or hops between JNs, and it only saves the information concerning the JNs that register with it.

As described in the previous section, JNs register themselves in the CRP and keep their information up-to-date by periodically refreshing it (sending JnUpdate messages). The CRP uses the timestamps in the JN entries to remove outdated entries in order to avoid working with possibly invalid JNs. This is achieved with a configurable JN entry freshness value.

2.5 Jumping through the Network

The proposed approach is materialized in the two part Jumper algorithm presented in Algorithms 1 and 2. Algorithm 1 depicts the integration of the decision algorithm

Algorithm 1 Jumper algorithm (in the routing protocol).

```
1: when routing message $m$
2:     JumperNode $jn \leftarrow$ getRandJumperNode( )
3:     Cell $best \leftarrow$ getBestCell($m.dst$)    ▷ regular routing protocol
4:     if $jn \neq null$ then
5:         send JumpReq[$best, m.dst$] to $jn$ (or CRP)
6:     else keep regular routing
7: when recv JumpRep[$jn_c$] from $jn$ (or from CRP)
8:     if $jn_c \neq null$ then
9:         send JumpMsg[$m, jn_c$] to $jn$ (or to $jn_c$)
10:    else keep regular routing
```

Algorithm 2 Jumper algorithm (in the CRP).

```
1: when recv JumpReq[$best, dst$] from $n_i$
2:     purgeExpiredNodes( )
3:     JumperNode $jn_c \leftarrow$ getBestJumperNode($best, dst$)
4:     send JumpRep[$jn_c$] to $n_i$
```

with the routing protocol at each node, while Algorithm 2 shows the decision logic residing in the CRP.

2.5.1 Contacting the CRP

During the routing process (e.g., in every hop or in a probabilistic manner), when a node receives a message to forward, it first checks its JN list for a neighboring JN and gets one at random (function getRandJumperNode at line 2 of Algorithm 1). The random selection is a very simple, but effective, load balancing strategy that guarantees[1] a uniform distribution of load among the JNs. However, other strategies may be used. For instance, we can add weights to the JNs and prefer the ones closer to the message destination.

Obviously, if the forwarding node is itself the chosen JN, it can contact the CRP directly. Otherwise, it will have to contact that chosen JN, which, in turn, will contact the CRP, in order to know how to proceed.

Naturally, analyzing the best path at every and each hop might incur in prohibitive latencies and degrade the performance of the routing protocol. Thus, we plan to make this into a configurable knob (to enable testing) and implement two main variations of this part of the algorithm: (i) one where only if the forwarding node is a JN it will execute Algorithm 1; and (ii) another where the JN is selected from a list of neighboring JNs. Both variations will be divided into two other variations. One where the best path analysis is done in every hop (respecting the previously defined variations (i) and (ii)) and another where the analysis is done in a probabilistic manner (e.g., tossing a coin). Naturally, in variation (ii), the chosen JN will work as a proxy of the forwarding nodes for contacting the CRP. These variations map into different implementations of the getRandJumperNode( ) function.

As part of the regular routing algorithm, when searching for the next hop, a forwarding node searches its neighbors for the best suitable cell to forward the message to (e.g., the closest neighboring cell to the message destination). This behavior is encapsulated in function getBestCell. If the forwarding node has a valid neighboring JN, before forwarding the message to the best next hop, it contacts the chosen JN (or the CRP directly, if it is the chosen JN itself), sending a message of the form

$$\text{JumpReq}[best, dst]$$

(where *best* is the best next hop chosen by the forwarding node, and *dst* is the message destination cell), and waits for the reply.

2.5.2 CRP's Decision Logic

Algorithm 2 captures the main decision logic for choosing the best JN (if it exists) to make the message "jump" through the network. When the CRP receives a JumpReq message, it first purges all the expired JNs from its list based on the update timestamps. Next, since we are using a geographic routing protocol, messages are addressed to concrete positions in space (in our case, cells), so it is possible to know the distance of each JN to the message destination. Thus, the best JN is chosen from all the valid ones, taking into account their distance to the message destination. The best JN, jn_c, is chosen in a way that

$$\nexists jn_i : jn_i \neq jn_c \wedge dist(jn_i.cId, dst) < dist(jn_c.cId, dst)$$

i.e., jn_c is the JN that minimizes the distance to the message destination. If no such JN exists, than *null* is returned.

Additionally, in order for the message jumps to pay off, the ratio between the cost of sending a message via the infrastructure versus ad-hoc networking should be greater than a configurable threshold, MIN_COST. To be able to decide between both paths, the CRP computes an estimate of the cost of sending the message by either one in terms of expended energy. Consider $\tau(n_s, n_d, m)$ a function that conveys the energy spent to transmit message m from source node n_s to destination node n_d, and $\iota(n, m)$ a function that denotes the energy required to transmit that same message from node n to (or from) the infrastructure. The cost of transmitting a message m between node n_s and n_d, via forwarding node n_f, in the ad-hoc network is, thus, given by

$$\mathcal{A}(n_s, n_f, n_d, m) = \tau(n_s, n_f, m) + \sum_{i=n_f}^{n_d} \tau(i, i+1, m)$$

while the cost of transmitting that same message resorting to a jump from node jn_s to jn_d is given by:

$$\mathcal{I}(n_s, n_d, jn_s, jn_d, m) = \sum_{i=n_s}^{jn_s} \tau(i, i+1, m) + \iota(jn_s, m) \\ + \iota(jn_d, m) + \sum_{i=jn_d}^{n_d} \tau(i, i+1, m)$$

Of course, these functions are heuristics can only work with estimated values for the expended energy when transmitting messages (that can depend on the used devices). Since we are using a geographic routing protocol, we can calculate a distance metric between nodes. Thus, functions $\mathcal{A}(n_s, n_f, n_d, m)$ and $\mathcal{I}(n_s, n_d, jn_s, jn_d, m)$ can try to estimate the number of cells (and, thus, the number of hops) messages will have to be routed through.

Assuming a starting and a destination jumper nodes, and the best next hop chosen by the forwarding node—denoted respectively by jn_s, jn_c, and *best*—we may now compute the ratio between the costs of sending a message from a node n_s to the destination ($m.dst$ in the algorithms), and decide that

$$\frac{\mathcal{A}(n_s, best, m.dst, m)}{\mathcal{I}(n_s, m.dst, jn_s, jn_c, m)} > \text{MIN\_COST} \implies \text{jumping}$$

[1] As much as possible, given that the set of JNs is not static.

The choice of a low MIN_COST threshold causes communication to be directed to the ad-hoc network only when the nodes are in very close cells or when no infrastructure is present at all. Conversely, a high value resorts to the infrastructure only when nodes are geographically far, being the threshold's value proportional to how actually *far* is far. Naturally, this threshold is a value that needs to be finetunned with some previous testing in the actual networks where this approach will operate in.

2.5.3 *CRP Reply*

After choosing the best JN, the CRP sends the reply back in the form of a

JumpRep[jn_c]

message. Using the JN in the reply message, jn_c, the forwarding node can make the message jump to that JN and continue its usual routing. Now, the emission of a

JumpMsg[m, jn_c]

message, where m is the message and jn_c is the target JN, follows the same procedure as for a JumpReq message. If the forwarding node is itself the JN, it can contact directly with the received JN, jn_c. Otherwise, the forwarding node has to work with the used JN and use it as a proxy, that will send the JumpMsg to the other JN, jn_c.

Another alternative is to use the CRP as the relay between the two JNs. But that will only pay off if the message size is not too big. For instance, if when sending the jumpReq message to the CRP we also send the message to be forwarded, the CRP will be able to forward the message to the chosen JN, if there is one. On the contrary, if there is no suitable JN, we spent energy and time sending the message to the CRP in vain. We plan to make this parameterizable using a message size threshold.

3. VALIDATION PLANNING

We plan to perform an extensive evaluation study (through simulation) to demonstrate the feasibility and validity of our ideas. To that end, we are currently implementing our proposed algorithm in the ns-3 simulator [15], and integrating it with the routing protocol. This evaluation study will allow for a better understanding of the pros and cons of this approach, and will enable us to measure the impact of our proposed routing protocol, while also enabling us to identify the key scenarios that maximize the benefits it may bring.

In this evaluation we will use the following metrics: (i) message drop rate; (ii) best path decision latency; (iii) amount of exchanged control traffic; (iv) end-to-end message latency; (v) overall network throughput; (vi) average energy spent by node; and (vii) total energy spent in the network. The best path decision latency and the amount of exchanged control traffic will allow to measure the overhead of our approach regarding the regular routing protocol.

We also plan to analyze the following variables: (i) network size; (ii) node density; (iii) node mobility; (iv) network diameter; and (v) amount of JNs. These variables will allow to understand and construct the scenarios that can take advantage of our approach.

The connectivity technology will also have some focus in the evaluation in order to build a better understanding in what is the impact of the different technologies (e.g., 3G/4G and Wi-Fi) on this approach.

4. RELATED WORK

There has been already some research in this topic, but the majority is related with the opposite idea, i.e., use the ad-hoc networks to offload network traffic from hot-spots in the infrastructure.

In [4], the authors propose the integration of infrastructure access with ad-hoc communications, allowing nodes to leverage on ad-hoc connections among them to alleviate the APs. This is achieved by dynamically switching between infrastructure and ad-hoc modes according to the instructions of the APs. In [5], the same authors a similar idea of a framework to establish ad-hoc connections between nodes in order to relay traffic from congested APs to non-congested ones.

Other works also tackle the combination of ad-hoc networks with infrastructure access by doing an analytical study around the transport capacity of ad-hoc networks with random topologies under the support of an infinite capacity infrastructure network [12]. Also, cellular-aided mobile ad-hoc network (CAMA) [3] proposes integrating ad-hoc networks with well-established cellular networks to improve communication and security. This is achieved by using out-of-band signaling and centralized control. In CAMA, control data goes through a cellular network, while all other data is kept in the ad-hoc network.

The work that more resembles our own is [7]. The authors present the notion of a vehicular grid as a large scale vehicular ad-hoc network and assume that, due to the ubiquitous presence of the infrastructure, APs exist in the areas vehicles flow. Whereas, we assume that infrastructure access might not exist at all and it is completely exterior to our approach, being out of our control. Nodes can have access to the infrastructure (either from 3G/4G networks or Wi-Fi APs) but it is only used as a rendezvous point and to store a small amount of network information.

5. CONCLUSION

In this paper, we argue for the opportunistic combination of ad-hoc network with infrastructure access as a way of allowing the usage of different paths for message routing. We propose a decision algorithm that determines the best path for routing messages during the forwarding process.

Currently, we are implementing the devised algorithm in the ns-3 simulator in order to do an extensive evaluation study. As future directions, we intend to add (configurable) weights to the different variables of the decision process of finding the best JN, and maybe add other variables like the nodes' battery level.

6. ACKNOWLEDGMENTS

This work was partially funded by FCT MCTES in the context of the research project CMUP-ERI/FIA/0048/2013, research grant SFRH/BD/99486/2014, and by NOVA LINCS UID/CEC/04516/2013.

7. REFERENCES

[1] F. Araujo, L. Rodrigues, J. Kaiser, C. Liu, and C. Mitidieri. Chr: A distributed hash table for wireless ad hoc networks. In *Proceedings of the International Workshop on Distributed Event-Based Systems*, pages 407–413. IEEE Computer Society, 2005.

[2] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic. *Mobile Ad Hoc Networking: The Cutting Edge Directions*. Wiley-IEEE Press, 2013.

[3] B. Bhargava et al. Integrating heterogeneous wireless technologies: A cellular aided mobile ad hoc network (CAMA). *Mob. Netw. Appl.*, 9(4):393–408, Aug. 2004.

[4] J. Chen, S. H. G. Chan, and S.-C. Liew. Mixed-mode wlan: the integration of ad hoc mode with wireless lan infrastructure. In *Proceedings of the Global Communications Conference*, volume 1, pages 231–235 Vol.1. IEEE, Dec 2003.

[5] J. Chen, J. He, and S.-H. G. Chan. A framework to relieve wireless hot-spot congestion by means of ad hoc connections. In *Proceedings of the International Conference on Mobile and Wireless Communications Networks*, pages 7–10. IEEE, 2003.

[6] U. Drolia, R. Martins, J. Tan, A. Chheda, M. Sanghavi, R. Gandhi, and P. Narasimhan. The case for mobile edge-clouds. In *Proceedings of the International Conference on Ubiquitous Intelligence & Computing*, pages 209–215. IEEE Computer Society, 2013.

[7] M. Gerla, B. Zhou, Y.-Z. Lee, F. Soldo, U. Lee, and G. Marfia. Vehicular grid communications: The role of the internet infrastructure. In *Proceedings of the 2Nd Annual International Workshop on Wireless Internet*, WICON '06, New York, NY, USA, 2006. ACM.

[8] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proceedings of the International Conference on Universal Personal Communications*, volume 2, pages 562–566 vol.2. IEEE, Oct 1997.

[9] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16(4):11–21, 2002.

[10] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings of the International Multi Topic Conference*, pages 62–68. IEEE, 2001.

[11] B. Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the International Conference on Mobile Computing and Networking*, pages 243–254. ACM, 2000.

[12] U. C. Kozat and L. Tassiulas. Throughput capacity of random ad hoc networks with infrastructure support. In *Proceedings of the International Conference on Mobile Computing and Networking*, pages 55–65. ACM, 2003.

[13] J. Li, J. Jannotti, D. S. De Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the International Conference on Mobile Computing and Networking*, pages 120–130. ACM, 2000.

[14] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. RFC 3561, RFC Editor, July 2003. http://www.rfc-editor.org/rfc/rfc3561.txt.

[15] G. F. Riley and T. R. Henderson. The ns-3 network simulator. In *Modeling and Tools for Network Simulation*, pages 15–34. Springer Berlin Heidelberg, 2010.

[16] J. Rodrigues, J. Silva, R. Martins, L. Lopes, F. Silva, U. Drolia, and P. Narasimhan. Benchmarking wireless protocols for feasibility in supporting crowdsourced mobile computing. In *Proceedings of the International Conference on Distributed Applications and Interoperable Systems*, pages 96–108, Cham, 2016. Springer International Publishing.

[17] The Serval project. http://www.servalproject.org, 2015. Accessed: September 2016.

[18] The span project. https://github.com/ProjectSPAN, 2015. Accessed: September 2016.

[19] J. A. Silva, R. Monteiro, H. Paulino, and J. M. Lourenço. Ephemeral data storage for networks of hand-held devices. In *Proceedings of the International Symposium on Parallel and Distributed Processing with Applications*. IEEE, 2016.

[20] A. Teófilo, D. Remédios, H. Paulino, and J. Lourenço. Group-to-group bidirectional wi-fi direct communication with two relay nodes. In *Proceedings of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 275–276. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015.