Opportunistic Routing in Dynamic Ad Hoc Networks: the OPRAH protocol

Cédric Westphal

I. INTRODUCTION

THE promiscuous nature of the air interface has been typically considered a hindrance to transmit more than an opportunity. Overhearing the other participants' traffic in the wireless communication network is assumed to be *interference,* an issue that should be remedied. Only recently has a different tack been taken, to use this fundamental broadcast characteristic of the air interface to the advantage of the communication network.

Ad hoc networks, where wireless nodes are within range of each other and use each other to relay messages seem an obvious place to attempt to leverage the broadcast nature of the air interface. However, typical ad hoc routing protocols seem more interested in finding one *good* route for the *whole life* of a connection than to make use of the diversity of paths available at any instant. The promiscuous nature of the air interface might be used at the route establishment time, to for instance determine potential back-up paths, but is mostly ignored during the transmission phase for current ad hoc routing protocols.

We will present in this document a protocol which uses the promiscuity of the air interface to find a more optimal path for each packet in a *dynamic* network. Before going more into the details, we want to emphasize a few points.

First, while cross-layer design offers means to optimize and enhance the protocol, we chose to focus on a strict layer separation, which allows us to re-use legacy physical and link layers. This is important for the application of the protocol in real systems, as it is backward compatible up to the routing stack. While cross-layer design is valuable, we will see that the link layer information, such as received signal strength indicator (RSSI) or channel quality indicator (CQI), will only give an idea of the best route at the time of the connection set-up anyway, and not throughout the life of a connection.

Second, there is a trade-off to assess between reliability and the number of transmissions. This take the form in our protocol of potential duplicate packets. We believe this can be addressed at the application layer –or that our protocol could be refined to handle duplicates– and that it is better to receive the packet in a timely manner at the cost of a few superfluous duplicates, than to not receive the packet at all.

Third, opportunistic protocols in general should not be applied to very power-scarce environments, as listening to the other nodes' transmissions, even if only at the MAC layer, has a cost in battery life. On the other hand, an 802.11 card for instance will listen to the transmissions on its channel and will have the corresponding power capacity. These are the types of nodes that we are considering. To phrase it differently, we are presenting a protocol which applies to some ad hoc and mesh networks but not necessarily to wireless sensor networks.

Before we describe the protocol, we would like to introduce some motivation for opportunistic protocols in general.

II. MOTIVATION AND RELATED WORK

We first consider a simple case: a square area. We wish to establish a connection between two nodes that sit at some diagonally opposite corners, which we denote s and d. The nodes are too far to connect directly, and need relays, which are uniformly distributed in the square. The connectivity between each node pair is affected by some fading. We will discuss the fading assumptions shortly.

Depending on the channel conditions, different paths may be available to relay the packets between s and d. The task of a routing protocol in such ad hoc network is to find a path which will allow the establishment of a reliable connection between s and d.

There are two broad types of routing protocols: reactive and proactive. A proactive protocol will try to identify the connectivity between the nodes ahead of time, and store the information in a route table somewhere, such that a route is available when needed. A reactive protocol attempts to identify the route only at the time of the connection establishment.

For now, we can consider static nodes, and look at the establishment of a *single* connection. In this set-up, both proactive and reactive protocol behave the same way: they attempt to identify a route that will last the length of the connection. If the route evolves during the time of the connection, both types of protocols have to handle this in some way.

In a static environment, where connectivity is perturbed only by fading between two fixed points, the performance of these protocols depends on how steady a route is. Experimental results by implementing and deploying an ad hoc networks have shown that most protocols fare poorly. For instance, [4] shows that existing ad hoc protocols do not perform well.

We ran some NS-2 simulations, using the Propagation/Shadowing module with typical outdoors parameters (with a path loss exponent $\beta = 2$ and a standard deviation $\sigma_{dB} = 5$) and the default 802.11 link layer model. We placed 20 nodes in a square area with a varying size ranging from 500x500, to 2,000x2,000. Again, we added a source s and a destination d in opposite corners and computed the length of the shortest path to cross the square diagonally.

We see on Figure 1 that there is a dramatic variation of the shortest path between s and d. We see that the shortest path is

---

C. Westphal is with the Networking Laboratory, Nokia Research Center, Mountain View, California. E-mail: cedric.westphal@nokia.com

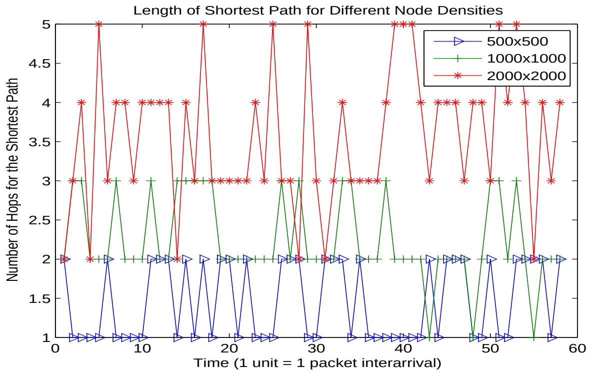

Fig. 1. Shortest Path in Number of Hops as a function of time

subject to some significant variations, even though we are considering a static network. Most current ad hoc protocols settle on a single route, the shortest path at the time of establishment for a reactive protocol, or the shortest path at the time of compilation of the route table, for a proactive one. Both prove harmful to the performance of the system: either the protocol settles for a shortest path that is not sustainable, or it settles for a path that is under-performing every time there exists a shorter path.

The benefit from opportunistic routing has not been fully tapped yet in the context of ad hoc networks. Opportunistic scheduling has been extensively used in one-to-many transmissions, mostly in scheduling nodes with respect to the conditions of the channel which connects them to a base station.

Cooperative strategies have been used in information theory, starting from Cover and El Gamal seminal paper [3] on the relay channel, and subsequent work in this community (see for instance [**?**], or the earlier survey in [8]).

In the ad hoc context, there are relevant works to consider which take advantage of the diversity offered by multiple users over the air interface. Leveraging the promiscuous nature of the air interface to define multiple paths in ad hoc network has been studied, for instance in [5] or [6]. [5] uses the fact that an AODV route request and reply is overheard by multiple nodes to build back-up routes in case the principal route fails. [6] computes multiple disjoint path to achieve the same purpose. As we will see below, the route set-up in OPRAH is very similar to these, the main difference being the use that OPRAH makes of the availability of multiple paths or relays during the connection.

The most relevant work on our topic is [1]. This work by Biswas and Morris, from MIT, came out of a practical problem: the RoofNet project [10] was first deployed using traditional and standardized MAC layers and network and routing protocols. However, these protocols turned out to perform somewhat poorly [4].

The idea in [1] is to use diversity from one sender to multiple receiver. To achieve this, they modified two key elements: the routing and the MAC protocol. The routing is modified so that there is not one single candidate for the next hop, but rather a list of possible candidates: all the nodes that would forward the packet closer to the destination (in the network topology sense) are included in the routing table.

The MAC is adapted to allow for different receivers to receive the same packet: the list of intended receivers for the next hop is included in the MAC header. The intended receivers acknowledge in turn, so that the sender knows that at least one receiver has received the packet successfully. All receivers are also assumed to overhear the acknowledgement to decide whether or not to forward their copy of the packet.

As we are interested in modifying only the routing layer, we cannot use the MAC of [1].Further, [1] is built upon a static network, so defining the set of potential candidates to forward a packet is relatively easy, but it becomes problematic in a dynaminc environment. We will define a protocol which copes with the changing environment and the mobility of the nodes, and is built on the use of standardized MAC, for instance IEEE 802.11.

III. The OPRAH protocol

Using the lessons we derived from the literature review, we are now able to define some guiding principles for the investigation of opportunistic routing in ad hoc networks.

A. Framework and Requirements

We consider an ad-hoc network composed of IP enabled modes. We consider that each node listen to the same channel for reception and transmission. This assumption is motivated by the use of an 802.11 MAC layer with single radio nodes. This simplification is welcome to allow nodes to talk to each other while in a dynamic environment: nodes do not have to agree on a dedicated channel for communication.

Each node relays traffic for the other nodes. We assume that all participants in the network perform their relaying duties. We specifically assume that nodes have no knowledge of their relative geographical positions.

Our goal is to define a routing protocol, and the associated MAC protocol such that:

- All nodes that could forward the packet to the destination and are located ”closer” than the sender should compete to forward the packet.
- The notion of ”closer” should be defined without the assistance of external measurement or probes (as is conducted in [2] for instance) but only with local information.
- The protocol should scale. As such, this prohibits the use of lists to replace or enhance routing tables. Reactive AODV-like mechanisms should be preferred.
- The protocol should be implementable using relatively simple off-the-shelf components.
- The protocol should be able to inter-operate with wired protocols, to ensure that the ad hoc network can communicate with the legacy wired infrastructure.
- The protocol should work in a low mobility environment first, as dealing with mobility will introduce new challenges.

There should be two phases: one is a route set-up phase. As node have no or little information in their route tables, a basic AODV route request would be issued. However, the protocol would diverge from AODV by trying not to find a single route to the destination, but by trying to fill in some distance information about the nodes up and down stream from the node.

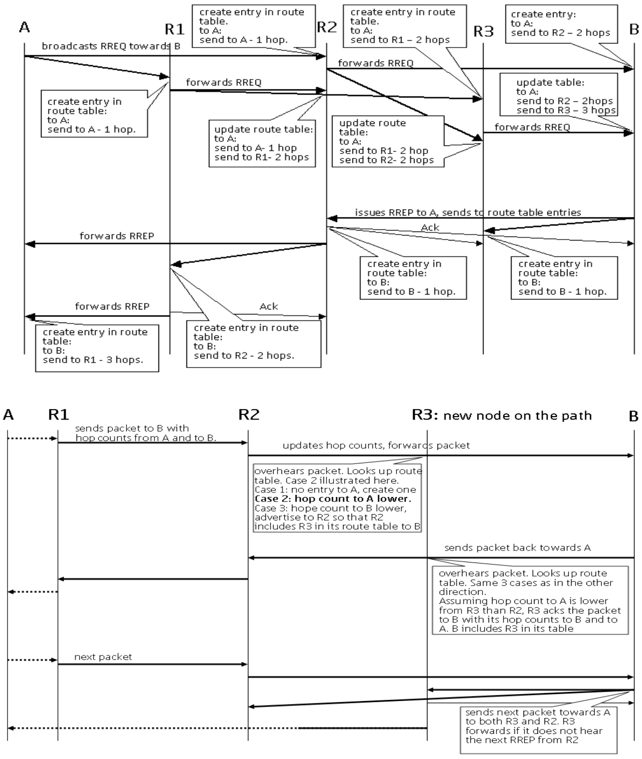

Fig. 2. Establishment (top) and Update (bottom) of the Route Entries for Opportunistic Routing Protocol

The key insight is that, for 802.11 MAC, a transmitter will request the channel at the exclusion of its neighbors. These neighbors are forced to idle until transmission ends. By using broadcast –albeit mostly one-hop, *local* broadcast– for all transmissions, the OPRAH protocol keeps these neighbors involved as potential relays.

The second phase is the forwarding phase per se. Using the list of potential relays, the packets get sent from the source to the destination. The promiscuity of the air interface can be used in the second phase to refine the list of potential relays at each node, and to adapt to the mobility and to the dynamic environment.

B. Protocol Description

We now formally describe the protocol. The discovery of relaying candidates is performed through backward learning, as for instance in [5].

A source S wishes to establish a connection to a destination D. It sends a broadcast message. The broadcast message is flooded. The format of the broadcast is that of a route request RREQ, similar to AODV [9]. Each node n which receives the RREQ for the first time, does three things:

- n increases by one the hop count h_S in the packet header.
- n forwards the updated packet to all its neighbors, by rebroadcasting it.
- n creates an entry for the route from n to S in its route table. An entry for a given destination is a list of (forwarding node; hop count to destination h_S) pair. n adds the pair (node it received the packet from; packet header hop count) as the forwarding node toward S.

Because of the nature of the flooding, a node n might receive several copies of the route request. When n receives another copy of the RREQ from S to D by way of n', it compares the hop count h in the packet header with the different hop count values stored in its route table. If the hop counts in the header is less than or equal to the minimal value stored in the route table for n's route to S, then n replaces its entry towards S in the route table with the new pair (n', h'_S).

One expects the value of the hop count in the duplicate packets to increase with time, so that the route table will accept new entries for a short period of time, then will ignore the next route requests. A duplicate route request is not broadcast again.

The RREQ establishes thus different paths back to the source. The destination D proceeds with the route request just as any other node, except obviously that it does not forward it further. D creates a route reply RREP, and sends it to all its neighboring nodes. The route reply is addressed to a broadcast address, with the actual destination S a parameter in the routing protocol header. The route reply also includes two values: one, h_D increases at every hop, and is used to assess the number of hops to route a packet from the node which receives the RREP back to D. The other, h_{SD} is the total number of hops from D to S, and is the lowest value of the hop count for the RREQ which reached D.

When a node n receives a route reply, it takes the following steps (keep in mind that the RREP now goes from D back to S):

- It checks if it has a route to D.
- If it does not have a route to D, it creates one with the hop count h_D.
- It checks if it has a route to S.
- If it does not have a route to S, it drops the RREP.
- If it does have a route to S, it computes the following $\delta = h_D + h_S - h_{SD}$. This is the difference from the path to the (so far) optimal route. If $\delta > \gamma$, where γ is some predefined threshold, the packet is dropped. This ensures that the routing protocol is free of infinite loops, as an infinite loop would increase h_D beyond the constant value $h_{SD} + \gamma$.
- If $\delta \leq \gamma$, then the node wait for a time proportional to δ, say $\delta\tau$.
- If during the waiting time, it receives another RREP from a node n' with $h'_D \leq h_D$, then it drops the RREP.
- Otherwise, it rebroadcasts the RREP to its neighbors.
- S may broadcast a packet reception message immediately using the RREP format with h_S set to 0, to prevent multiple nodes to forward the RREP on the last hop.

A node n which receives the route reply from node n' updates its route table.

- it increases the hop count h'_D to D by one. $h_D \leftarrow h'_D + 1$.
- it insert its minimal value h_S for the number of hops towards S in the header.
- it adds the pair (n', h'_D) as an entry in the list of potential forwarding node towards D.

When the packet (or potentially the multiple packets) arrive back at the source S, the routing has been established in both directions. The connection can then starts, with the rule to forward the packets similar as the rules for the RREP in both direction. Each packet being sent reinforce and potentially updates the information in the route tables, as new nodes might be added by overhearing the broadcasts and responding to the packet.

All entries in the route table are kept for the time of the connection only. They should be set in the table with a default TTL value of the order of a few round trip times. There is no need to keep the information longer in highly dynamic environment, and this ensures the scalability of the protocol.

Figure 2 describes a simple case to illustrate the protocol. A node A wishes to establish a connection to a node B, with relays R1, R2, and R3 potentially available in between.

The forwarding of packets after the connection is established is done using the route tables as indicated on the figure. If some transmissions failed the first time around, then the route table is updated after the next successful transmission. For instance, in Figure 2, the RREP between R2 and A is dropped, prompting R1 to send the RREP again. The next packet that R2 send to A might go through, at which point A includes an entry to R2 towards B in its route table.

The bottom of Figure 2 describes how the route entries are updated during a connection. This allows to deal with some level of mobility, or to deal with slightly outdated routes, ie. routes that are still working but not optimal: a better route will insert itself in the route tables automatically.

The evaluation of the proposed protocol is presented in the next section.

IV. PROTOCOL EVALUATION

We conducted the simulations in a 500x500 square area, and again trying to connect two nodes situated at opposite corners of the square via N nodes uniformly distributed in the square. We first picked $N = 75$ and looked at the performance of the protocol using the GE model described in II.

In the static case, the AODV protocol converges on a route that is stable throughout the simulation, as the nodes do not move. On the other hand, it converges towards a route that is sub-optimal. In Figure **??**, we plot the ratio of the path length for each packet for OPRAH divided by the path length for AODV for different realization.

OPRAH consistently finds a shorter path which is on average 77% of the length of the AODV path. This is significant enough, especially since the network is static. Furthermore, the performance of AODV also involves repairing the route after it breaks, an extra overhead which occurs less frequently with OPRAH.

Figure 3 performs the same measurement using the correlated Rayleigh fading channel model instead of the GE channel model. We see similar gain, with a path length for OPRAH routes about three quarters on average that of AODV.

To see the impact of the route repairs, we consider a dynamic network in which nodes move about. In this scenario, we consider a fixed connectivity radius, and the variation in connectivity is only due to the movement of the nodes. In our simulation, $N = 50$ and the connectivity radius is 250m. Our mobility model is as follows: every node picks a speed and a direction. When it hits the boundary, it picks another speed and direction so that it stays within the square. Speed and directions are picked uniformly. The speed is bounded below by a small minimal value so that no node stays stuck in a low speed for ever.

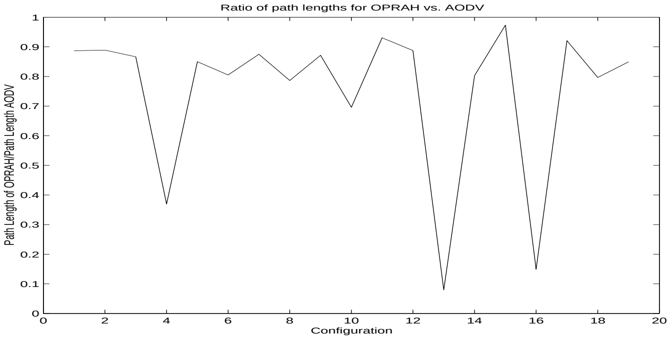

Fig. 3. Length of path in static network, correlated Rayleigh model

OPRAH is relatively robust to node mobility. We plot on figure 4 the path length achieved by the OPRAH protocol as a function of the mobility of the node. Node mobility was generated in NS-2 using the Random Waypoint Model with pause time set to zero, and the speed bounded below by 0.1 m/s. We placed 20 nodes in a square area of 1,000x1,000. Irrespective of the node mobility, each packet is forwarded in about two hops.

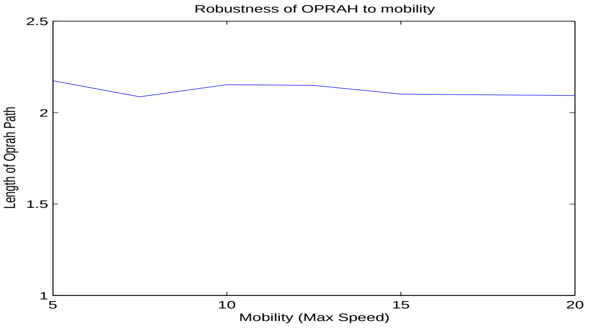

Fig. 4. Path Choice as a function of speed

REFERENCES

[1] S. Biswas, R. Morris, *Opportunistic Routing in Multi-Hop Wireless Networks*, Proceedings of the Second Workshop on Hot Topics in Networking (HotNets-II), Cambridge, Massachusetts, November 2003.

[2] S. Biswas, R. Morris, *ExOR: Opportunistic Multi-Hop Routing for Wireless Networks*, Proceedings of ACM Sigcomm 2005.

[3] T.M. Cover, A.A. El Gamal, *Capacity Theorems for the Relay Channel* IEEE Trans. Info. Theory, vol.25, pp.572-84, Sept. 1979.

[4] D. De Couto, D. Aguayo,J. Bicket, R. Morris, *A High-Throughput Path Metric for Multi-Hop Wireless Routing*, Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03), San Diego, California, September 2003.

[5] S.-J. Lee, M. Gerla, *AODV-BR: Backup Routing in Ad Hoc Networks* in Proc. WCNC 2000.

[6] K. Marina, S. Das *On-demand Multipath Distance Vector Routing in Ad Hoc Networks*, in Proceedings of IEEE ICNP, November 2001.

[7] B. Milner, A. James, *An Analysis of Packet Loss Models for Distributed Speech Recognition* in Proc. ICSLP 2004, October 2004.

[8] A. Nosratinia, T. Hunter, A. Hedayat, *Cooperative Communication in Wireless Networks*, IEEE Communications Magazine, Oct. 2004.

[9] C. Perkins, E. Royer, *Ad hoc On-Demand Distance Vector Routing.* Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.

[10] *RoofNet project*, http://www.pdos.lcs.mit.edu/roofnet/