Least-Cost Opportunistic Routing

Henri Dubois-Ferriere Matthias Grossglauser Martin Vetterli
School of Computer and Communication Sciences
EPFL
1015 Lausanne, Switzerland

ABSTRACT

In opportunistic routing, each node maintains a group of candidate relays to reach a particular destination, and transmits packets to *any* node in this group. If a single candidate relay receives the packet, it becomes the effective relay to forward the packet further. If no candidate receives the packet, then the current sender re-transmits. If multiple candidates receive the packet, then the link layer chooses a single receiver to be the relay. This choice could be made at random, or it could be driven by information coming from the routing layer, for example to use the best receiver as the relay.

This paper addresses the *least-cost opportunistic routing* (LCOR) problem: how to assign and prioritize the set of candidate relays at each node for a given destination such that the expected cost of forwarding a packet to the destination is minimized. We solve this problem with a distributed algorithm that provably computes the optimal assignment of candidate relays that each node should allow to reach a particular destination. Prior proposals based on single-path routing metrics or geographic coordinates do not explicitly consider this tradeoff, and as a result make choices which are not always optimal.

1. INTRODUCTION

In wireless networks, it is often less costly to transmit a packet to *any* node in a set of neighbors than to one specific neighbor, using a link-layer primitive known as "anycasting". For example, with unreliable wireless links, the probability of a packet being successfully received by at least one node in a set of neighbors is usually greater than the probability of one specific node receiving it. This observation motivates the idea of *opportunistic* routing (OR) [1–4]. In OR, the next-hop relay decision is made *after* a packet has been transmitted, allowing a sender to opportunistically take advantage of outcomes that are inherently random and unpredictable. A key question is then to decide, at each node, which neighbors should be candidate relays to reach a destination, and how to prioritize and select the effective relay when multiple candidates have received a transmission.

Previous work has focused on mechanisms for link-layer anycasting, and on devising robust, low-overhead coordination protocols for receivers of a packet to agree upon a next-hop relay [1–6] when multiple candidates receive a packet. At the same time, comparatively little attention has been given to the problem of how to best select and prioritize candidate relays so as to minimize end-to-end forwarding costs.

The starting point of this work is the question: with OR, are there practical and general ways to a) compute the optimal candidate relays that can be used at each node to reach a given destination, and b) prioritize these relays in order to optimally select the effective forwarder when multiple candidates have received a packet? Of course, the notion of optimality is here relative to the model of a network, and any routing algorithm can only be as good as the model and input metrics that drive it. This point is particularly relevant in the context of wireless networks, where link statistics are hard to estimate and often must be paired with simplifying assumptions (e.g., independence).

The optimal selection of candidate relays must take into account the following tradeoff. On the one hand, taking many candidate relays often decreases the forwarding cost (i.e., the cost to send to *any* of these candidates). On the other hand, each neighbor does not make as much progress as the next hop in the shortest path to the destination. Therefore employing too many candidates may increase the likelihood of a packet veering away from the shortest route (and ultimately even introduce loops in the routing topology).

To solve the problem of finding optimal candidate relay sets and prioritizing the candidate relays, we use a generalization of single-path routing, where the next hop to reach a destination is explicitly treated as a *set* of neighbors rather than a single neighbor. The notion of single-path route is generalized to that of *opportunistic route*, which is the union of all possible packet trajectories induced by an assignment of candidate relays. Within this framework, we formulate a distributed algorithm for *least-cost opportunistic routing (LCOR)*, that computes the optimal choices of candidate relays. The LCOR algorithm is operationally similar to the classi-

Notation and Acronyms	
$N(i)$	Neighbors of node i
p_{ij}	Packet reception prob. from i to j
$J(i)$ (or J)	Candidate relay set (CRS) at node i
d_{iJ}	Anycast link cost (ALC) from i to J
R_{iJ}	Remaining path cost (RPC) from J to the destination

cal distributed Bellman-Ford, but is driven by different metrics that generalize unicast link and path costs respectively.

The rest of this paper proceeds as follows. Section 2 defines and motivates LCOR, Section 3 introduces the LCOR algorithm, and Section 4 gives properties and some insights into least-cost opportunistic routes. Section 5 describes related work, and Section 6 concludes.

2. PROBLEM OUTLINE

This section defines the least-cost opportunistic routing (LCOR) problem. The underlying communication primitive used by opportunistic routing (OR) is link-layer **anycast**, whereby a node transmits a packet to *any* node among a set of its neighbors. We call this set the **candidate relay set** (CRS), denoted $J(i)$ (or J, when i can be omitted without ambiguity); it contains all the nodes which may be used as next-hop relays for packets forwarded by i toward the destination[1].

With anycast transmission, a packet may travel according to a number of different paths from a source to a destination. We call **opportunistic route** (opp. route) the union of all possible paths between a source and destination, arising from a given assignment of CRS at each node. An opportunistic route $\mathcal{R}$ from a source to a destination is an acyclic directed graph where every node (but the source) is a successor of the source, and every node (but the destination) is a predecessor of the destination. Figure 1 shows an example of an opp. route. Each opp. route can be specified equivalently by the list of CRS $J(n_1), J(n_2), \ldots J(n_k)$ of the nodes $n_1, n_2, \ldots n_k$ it contains, or by the list of paths that can be used to traverse it.

2.1 Cost of opportunistic routes

2.1.1 Anycast link cost

In single-path routing, the overall cost of a route is the sum of underlying costs of the unicast links it traverses. With OR, we must generalize the notion of link cost from single-path routing, to account for anycast forwarding rather than unicast. We define the **anycast link cost (ALC)** d_{iJ} as the cost to send a packet from i to *any* node in the set J, where $J \subseteq N(i)$ is a subset of i's neighbors. As an example of ALC, we can generalize the expected transmission count (ETX) [7] metric for unicast transmission[2]. This metric counts the expected number of transmissions to successfully deliver a packet across an unreliable unicast link. With link-layer anycast, the ETX becomes the expected number of transmissions until any node in J receives the packet. Its expression is:

$$d_{iJ}^{ETX} = \frac{1}{p_{iJ}}, \qquad (1)$$

where p_{iJ} is the probability that a packet from i is received by *at least one* node in the set of nodes J:

$$p_{iJ} = 1 - \prod_{j \in J} (1 - p_{ij}). \qquad (2)$$

Note that this metric generalizes the unicast ETX, that is, for a singleton CRS with $|J| = 1$, the anycast ETX reduces to the unicast expression. Of course, it assumes spatial independence, such that i's transmission is received independently (or not) by node in J. This assumption is reasonable when fading and noise are the main source of channel errors; it may not hold when interference from other transmissions is a frequent source of errors. While our choice of metric is intentionally simple, other metrics that capture spatial loss correlations could be used in the LCOR framework.

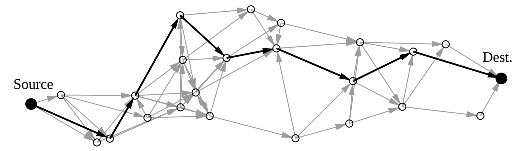


Figure 1: An opportunistic route is the union of all possible paths from a source to a destination that are induced by a given choice of candidate relays at each node. Each node has arrows pointing to its candidate relays to reach the destination, and a possible trajectory through the opp. route is highlighted in bold.

2.1.2 Cost of a trajectory in an anypath route

A trajectory T in an opportunistic route $\mathcal{R}$ is a sequence of nodes $(s, n_1, n_2, \ldots n_k, 1)$ between a source s and the destination 1 such that each of the pairs $(s, n_1), (n_1, n_2), \ldots (n_k, 1)$ are links in $\mathcal{R}$. In other words, a trajectory is a possible path that a packet can take across an opp. route. We now define the cost of a trajectory relative to the opp. route it traverses.

DEFINITION 1. *Let $T = (s, n_1, n_2, \ldots n_k, 1)$ be a trajectory in $\mathcal{R}$. The* cost *of T relative to $\mathcal{R}$, denoted $c(T|\mathcal{R})$, is the sum of the anycast link costs in $\mathcal{R}$ of the*

[1] In the remainder of this paper, it shall be implicit when referring to a CRS that it is relative to one particular destination, which can be any node in the network.

[2] Other examples of anycast link costs are developed in [8].

nodes in T:

$$c(T|\mathcal{R}) = \sum_{i \in T} d_{iJ(i)} = d_{sJ(s)} + d_{n_1 J(n_1)} + \ldots d_{n_k J(n_k)}.$$

It is important to emphasize that the cost of a trajectory depends on the opportunistic route $\mathcal{R}$ that it traverses, because each constituent ALC d_{iJ} depends on the entire candidate relay set J, and not just on the effective relay in J that is used. We illustrate this dependence in Figure 2, by computing the cost of the same trajectory $T = (a, b, c, d)$ relative to four traversed opportunistic routes. All links have delivery probability 0.5, and the ALC metric is d^{ETX}. In Figure 2(a), node a has two candidate relays, and so its ALC is $d^{ETX}_{aJ(a)} = (1 - 0.5^2)^{-1} = 4/3$. Nodes b and c have a single candidate relay and have ALC equal to 2, giving a path cost $c(T|\mathcal{R}) = 5.33$. In Fig. 2(b) the costs at nodes b and c are lower due to their additional candidate relays. In Fig. 2(c), the trajectory cost is the same as in (a), even though the opp. routes are different, because anycast link costs of nodes b and c are not changed by additional incoming links. Finally in Fig. 2(d), the opp. route and the trajectory are identical, with cost equal to the cost of the single-path route from a to d.

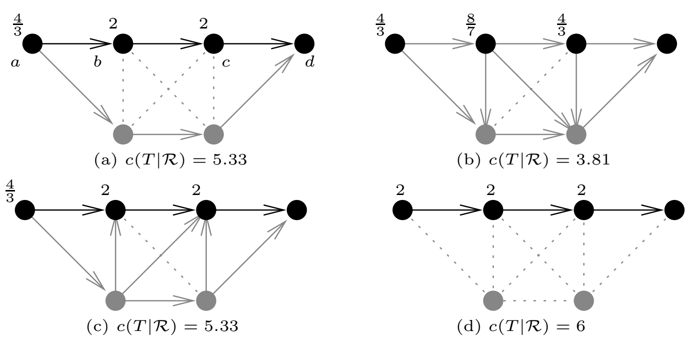


Figure 2: Cost of the same trajectory $T = (a, b, c, d)$ traversing four different opportunistic routes. The cost $d^{ETX}_{iJ(i)}$ is annotated next to nodes a, b, and c.

2.1.3 Least-cost opportunistic route

There are multiple possible trajectories to traverse an opp. route, and each is used with some probability $P(T)$, that depends on a number of factors, such as the non-deterministic outcome of link-layer transmissions, decisions made by link- and network-layer protocol mechanisms, and the topology of the network. It is then natural to define the cost of an opp. route as the expected cost of traversing it:

Definition 2. *The cost $C(\mathcal{R})$ of an opp. route $\mathcal{R}$ is the expected cost of all trajectories across that route,*

$$C(\mathcal{R}) = \sum_{T \in \mathcal{R}} P(T) \cdot c(T|\mathcal{R}),$$

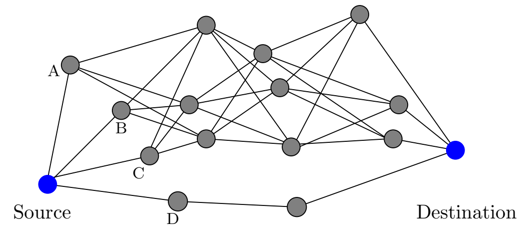


Figure 3: Mis-match of single-path metrics with opportunistic routing. Sending a packet via the dense mesh takes advantage of anycast forwarding and is often cheaper than via the four-node strand at bottom, even if it goes through more hops. However, the use of a single-path metric prevents the source from using any of its neighbors in the upper dense area, because in single-path distance they are further from the destination than the source itself.

where the sum is over all *possible trajectories from the source to the destination of $\mathcal{R}$.*

This definition generalizes the cost of a single-path route: if all CRS's are singletons, there is only one trajectory T across an opp. route (and so $\mathrm{P}(T) = 1$), and its cost is the sum of its constituent link costs. Each possible choice of candidate relays gives rise to a probability distribution over all possible paths between the source and destination, and this distribution determines the expected cost of using a route. Fortunately, we shall see in Section 3 that it is possible to avoid the explicit computation of this distribution.

Having now defined opp. routes and their cost, it is natural to define the least-cost opportunistic route ("LCOR route") between two nodes as the one with minimal cost:

Definition 3. *The least-cost opportunistic route (LCOR route) $\mathcal{R}^*$ from a source to a destination is the opp. route that has lowest cost $C(\mathcal{R}^*)$ of all opp. routes between those nodes.*

As a direct consequence of this definition, the LCOR route has cost either smaller than or equal to the shortest single-path cost between two nodes, since the set of all opportunistic routes between two nodes *includes* the set of single-path routes between these nodes. Note that there may be multiple LCOR routes with equal minimal cost (as is the case with single-path routes). Also, the least-cost opportunistic route may itself be a single-path route. For example, if the metric is ETX and all links have delivery probability 1, then the LCOR route is identical to the shortest single-path route.

2.2 Why not use shortest single-path metrics?

Certain existing opportunistic routing protocols are driven by single-path metrics: nodes run a single-path routing algorithm and choose candidate relays using a criterion that is based on the shortest-path distance of their neighbors to the destination. For example, a node running ExOR [3] takes as candidate relays all neighbors with lower single-path cost to reach the destination. Before developing our solution to the LCOR problem, we show why strategies based on shortest-path metrics do not always lead to optimal CRS choices.

Figure 3 shows a network where the source has four neighbors and must select a subset of these neighbors as the set of candidate relays that may be used to reach the destination. Let us assume that all links have packet delivery probability $p = 0.75$, and compute delivery probabilities using a single-path metric. The probability of a packet being successfully delivered to the destination when sending via D through the two-node strand at the bottom is $p^3 = 0.42$. The probability of a packet being successfully delivered when going through any 4-node path in the mesh at the top is $p^5 = 0.24$. A single-path metric would therefore lead us to select node D as the sole candidate relay from the source, since A, B, and C each have a lower delivery probability to the destination than the source itself. However, with anycast forwarding, each node in the upper mesh has three candidate relays to its right, and so the probability of delivery across the upper mesh is actually higher than 0.24. Indeed, a simple computation shows that the true delivery probability, when using A, B, C as candidate relays and going through the upper mesh is $(1-(1-p)^3)^4 \cdot p = 0.70$. If our choice of candidates is driven by single-path metrics, we would ignore this opportunity, and as a result make a routing decision that provides a significantly lower delivery probability; the single-path metric effectively *disqualifies* nodes that in fact should be candidates.

3. FINDING LEAST-COST OPPORTUNISTIC ROUTES

While the definition of opportunistic route cost (Def. 2) is intuitive, it sheds no light on how to actually *compute* this cost in a distributed setting, let alone how to find the opportunistic route with least cost. This section introduces a distributed algorithm to compute the optimal candidate relay sets, based on Bellman-Ford.

3.1 Remaining path cost

With unicast forwarding, it is trivial that the remaining cost for a packet to reach the destination after it is forwarded to the relay is the path cost from the relay to the destination. With anycast forwarding, the effective relay can be any node in J, and so the corresponding notion must be revisited. We define the **remaining path cost (RPC)**, denoted R_{iJ}, as the expected cost to reach the destination from the CRS J to which node i has anycast a packet. The breakdown of an opportunistic route's cost into ALC and RPC is illustrated in Figure 4. Like for the anycast link cost, establishing

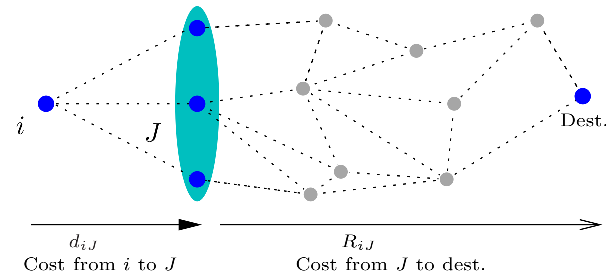


Figure 4: The cost of an opportunistic route can be separated into two components: the anycast link cost, which is the cost to reach the next-hop relay, and the remaining path cost, which is the cost to get from the next-hop relay to the destination.

the RPC is a modelling decision, and its expression can differ for various instantiations of LCOR.

This notion of a distance from a *set* of nodes J to the destination may appear somewhat disconcerting. The key is to note that the RPC is a weighted combination of costs from each node in J to the destination. The weights reflect the relative probability that each node in J is effectively used as relay and forwards a packet that was link-layer anycast from i to J.

As an example of RPC, consider an ideal anycast link layer operating as follows. The sender i transmits a packet. If a single node in J receives the packet, that node is used as the relay. If multiple nodes in J receive the packet, then the receiver with lowest cost to reach the destination is selected as the relay. If the packet is not received by any node in J, the sender retransmits. The behavior of non-ideal, practical link layers can also be captured in the RPC and is further discussed in Section 4.

Denote by D_k the cost to reach the destination from a node k. If $D_k = D$ for all $k \in J$, then the RPC with our ideal link layer is simply $R_{iJ} = D$. Now consider the case where all D_k are not equal, but all link delivery probabilities are equal to some p. In this case, the RPC can be computed as

$$R_{iJ} = \frac{p}{1-(1-p)^n} \sum_{j=1}^{n} (1-p)^{j-1} D_j, \qquad (3)$$

where it is assumed (without loss of generality) that the nodes in J are sorted by their cost to the destination, i.e., that $D_1 < D_2 < \ldots < D_j$. Finally, in the general case each node k in J receives the packet with some probability p_{ik}. The remaining path cost is then:

$$R_{iJ} = \frac{1}{1-\prod_{k \in J} \overline{p_{ik}}} \left(p_{i1} D_1 + \sum_{j=2}^{n} p_{ij} D_j \left(\prod_{k=1}^{j-1} \overline{p_{ik}} \right) \right). \qquad (4)$$

Note that like the anycast link cost, the RPC generalizes the single-path case: when $|J| = 1$, it simply becomes the cost from the relay to the destination. Note however that the RPC from a CRS J to the destination depends not only on J itself, but also on the predecessor node i of J. In other words, the same CRS J can give a different RPC for two different senders i. In this sense RPC is quite different to the single-path notion that it generalizes, since the traditional single-path notion of path cost does not depend on any *previous* node in a path.

We illustrate the dependence of the RPC on the sender with the example of Figure 5. Two forwarding nodes i and j each have the same candidate relay set $C = C(i) = C(j) = \{k, l\}$. We consider the ideal link-layer described earlier, that selects as relay the receiver with lowest cost to reach the destination. So k is chosen as relay every time it receives a packet (because $D_k < D_l$), and node l is only chosen if it receives a packet and k does not. What are the remaining path costs R_{iC} and R_{jC} to the destination? Consider first sender i. Node k receives every packet from i, and so is the effective relay for every packet (including for packets that node l had received as well). The remaining path cost R_{iC} is thus 5. The situation is different for sender j. Node k only receives packets from j with probability 0.8, whereas node l receives every packet, thus $R_{jC} = 0.8 * 5 + 0.2 * 10 = 6$. Note that the ideal link-layer is not required to illustrate $R_{iC} \neq R_{jC}$, e.g. the inequality would also hold if the relay is selected at random each time both k and l receive a transmission.

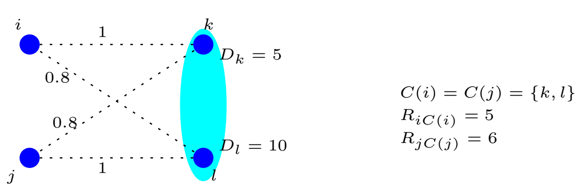


Figure 5: The remaining path cost depends not only on the CRS J but also on the sender i.

3.2 Physical cost criterion

While the ALC and RPC metrics can be designed in many different ways depending on the underlying protocol and cost model, they must jointly satisfy one condition in order for routing to converge. This is the physical cost criterion; it requires that if a node i adds to its CRS a neighbor with higher cost to the destination than i itself, then i's cost to reach the destination must increase. The physical cost criterion can be seen as a generalization of the requirement that link costs be non-negative in order for single-path routing algorithms to converge.

Definition 4. *Consider a node i with CRS J. The cost to reach the destination from i is $D_i = d_{iJ} + R_{iJ}$. Let $k \in N(i) \setminus J$ be a neighbor of i that is not in J, and for which $D_k \geq D_i$, and define $J' = J \cup k$. The physical cost criterion is respected if and only if:*

$$d_{iJ'} + R_{iJ'} \geq d_{iJ} + R_{iJ},$$

for all possible combinations of i, J, and k.

3.3 Least-cost opportunistic routing algorithm

How does a node select which of its neighbors should be candidate relay nodes? As illustrated in Fig. 4, the expression to minimize is the sum of the ALC and RPC, which must be minimized over all possible subsets $J \subseteq N(i)$:

$$D_i = \min_{J \in 2^{N(i)}} [d_{iJ} + R_{iJ}]. \quad (5)$$

This equation represents the steady-state of the LCOR algorithm, that computes least-cost opp. routes as follows. In one iteration, each node i updates its value D_i^h, where h is the iteration index. This D_i^h is the opportunistic routing cost estimate from i to the destination at the h-th iteration; it converges toward D_i. By convention, we take:

$$D_1^h = 0, \qquad \text{for all } h, \quad (6)$$

and we set $d_{ij} = \infty$ if (i, j) is not an link of the graph. One iteration step consists of updating the estimated cost to the destination from each node:

$$D_i^{h+1} = \min_{J \in 2^{N(i)}} [d_{iJ} + R_{iJ}^h] \qquad \text{for all } i \neq 1, \quad (7)$$

where R_{iJ}^h is the remaining path cost computed using the costs D_j^h, $j \in J$ from the previous iteration. The CRS used by i is found as a by-product of minimizing the above equation. Our definition of the algorithm is completed by noting the initial conditions:

$$D_i^0 = \infty, \qquad \text{for all } i \neq 1.$$

The algorithm terminates when:

$$D_i^h = D_{i-1}^h, \qquad \text{for all } i.$$

In the following, a $(\leq h)$ opportunistic route is one whose longest path contains at most h hops. A least-cost $(\leq h)$ opp. route from a node i is a least-cost opp. route from i to the destination, subject to the constraint that the longest path in the opp. route traverses at most h hops.

Proposition 1. *The LCOR algorithm computes, at iteration h, the least-cost $(\leq h)$ opportunistic route costs from each node to the destination. Furthermore, the algorithm terminates after at most $h^* \leq |\mathcal{N}|$ iterations, and at termination, $D_i^{h^*}$ is the cost of the least-cost opp. route from i to the destination.*

The proof of this proposition is given in [8]. The LCOR algorithm resembles the classical Bellman-Ford algorithm, with the crucial difference that the cost metrics are generalized to handle candidate relay sets rather than single relay nodes. Just like single-path Bellman-Ford, the algorithm works in a distributed setting, with nodes asynchronously recomputing their cost (using eq. (7)) and advertising it to their neighbors.

3.4 Reducing the Search Space

While the upper bound on the LCOR algorithm's convergence time (in number of iterations) is the same as for single-path Bellman-Ford, its complexity is greater, since there are $2^{|N(i)|}$ possible subsets that must be evaluated, compared to $|N(i)|$ possible relays with single-path routing. We now discuss some possible ways to reduce the complexity of computing (7).

3.4.1 Distance filtering

A first reduction comes under the assumption of a physical cost model. With such a model, we can do distance filtering and reduce the size of the set $N(i)$. Define the set of nodes $\check{N}(i) \subseteq N(i)$ consisting of all the neighbors k of i for which $D_k < D_i$. Then, we have that

$$\min_{J \in 2^{\check{N}(i)}} d_{iJ} + R_{iJ} = \min_{J \in 2^{N(i)}} [d_{iJ} + R_{iJ}],$$

where the proof follows immediately from the definition of the physical cost model that says that adding any node in $N(i) \setminus \check{N}(i)$ to the candidate relay set can only increase the distance D_i. So, we can take only the neighbors with lower distance to the destination than ourselves $D_k < D_i$ and perform the Bellman minimization over the nodes in this set. Of course, this simplification is helpful in practice, but it does not modify the exponential nature of the solution space size.

3.4.2 Subset ordering

While in the general case we must resort to heuristics to reduce the exponential size of the solution space, we are fortunate that the Bellman equation simplifies considerably, under a set of conditions that hold for some cases of practical interest. These conditions essentially require that there be an ordering on all the 2^n possible subsets of $N(i)$, that this ordering depend only on the D_i, and that the ordering be the same as the ordering of candidate relay sets that results from the costs $d_{iJ}+R_{iJ}$ of (5).

The first condition ($\mathbf{H_1}$) states that the remaining path cost decreases when we *add* a node to the candidate relay set with distance to the destination inferior than that of all nodes already in the set. The second condition ($\mathbf{H_2}$) states that the anycast link cost depends only on the size of the candidate relay set, and that this cost decreases as the set grows. The third condition ($\mathbf{H_3}$) states that the remaining path cost decreases when we *replace* a node in the CRS with another node having lower distance to the destination. These conditions are formally defined hereafter:

$\mathbf{H_1}$: For all $k \notin J$ and $J' = J \cup \{k\}$,

$$R_{iJ'} < R_{iJ} \text{ if } D_k < D_j \text{ for all } j \in J.$$

$\mathbf{H_2}$: For all J and J',

$$d_{iJ'} < d_{iJ} \text{ iff } |J'| > |J|, \text{ and}$$
$$d_{iJ'} = d_{iJ} \text{ iff } |J'| = |J|.$$

$\mathbf{H_3}$: For all $l, k \in J$ s.t. $D_l < D_k$,

$$R_{iJ'} < R_{iJ}, \text{ where } J' = J \setminus k \cup \{l\}.$$

Assume now w.l.o.g. that the n nodes in $N(i)$ are sorted by their distance to the destination, i.e., that $D_1 < D_2 < \ldots < D_n$.

Proposition 2. *In a network model satisfying* $\mathbf{H_1}$, $\mathbf{H_2}$, *and* $\mathbf{H_3}$, *the candidate relay set* $C(i)$ *minimizing the Bellman equation* (5) *is of the form* $\{1, 2, \ldots, j\}$, *where* $1 \leq j \leq |N(i)|$.

The proof of this proposition is given in [8].

Corollary 1. *In a network model satisfying* $\mathbf{H_1}$, $\mathbf{H_2}$, *and* $\mathbf{H_3}$, *finding the* $C(i)$ *that minimizes the Bellman equation* (5) *requires only searching through* n *possible sets.*

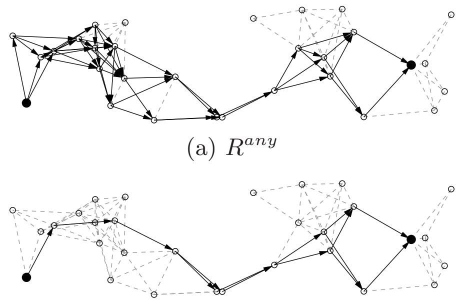
4. PROPERTIES AND INSIGHTS

This section uses the framework and algorithm of the previous sections to shed some insight on the interplay between LCOR, the underlying link-layer coordination protocol, and the cost and characteristics of least-cost routes.

4.1 Other policies for effective relay selection

When a packet transmitted by a node i is received by more than one node in i's CRS, a decision must be made as to which receiver should then forward the packet further. We call this an effective relay selection (ERS) policy. The previous sections assumed a policy that always chose as the next forwarding node the "best-placed" receiver, that is, the receiver k with minimum cost to the destination D_k. We call this policy *ERS-best*. The framework and algorithm outlined in the previous sections allow to model and capture the effect of other relay selection policies. Another example of ERS policy is *ERS-any*, where the relay is chosen uniformly at random among receivers of a packet.

In comparison with ERS-best, ERS-any has the disadvantage that it may select as relay a receiver with a more costly path to the destination than the least-cost receiver. At the same time, executing ERS-any in



(a) R^{any}

(b) R^{dup}

Figure 6: Comparison of least-cost opportunistic routes with perfect link layer coordination vs a link layer that sometimes lets through duplicate transmissions. In (b), the optimal LCOR route has smaller CRS's, because the reduction in forwarding cost from using large CRS is offset by cost of possible duplicate transmissions, in particular at large distances from the destination.

a protocol may have lower cost than executing ERS-best. Also, using ERS-any spreads the forwarding load more evenly than ERS-best over the entire opportunistic route. These arguments are qualitative. We do not seek to claim that ERS-any should be used over ERS-best, but rather to point out that other policies exist, and show how they can be modelled within LCOR.

With ERS-any, if $S \subseteq J$ is the set of nodes that receives a transmission, then the remaining path cost is the average cost over the nodes in S. The remaining path cost R_{iJ}^{any} can thus be written as

$$R_{iJ}^{any} = \sum_{S \in 2^{C(i)}} P(S) \left(\frac{1}{|S|} \sum_{j \in S} D_j \right), \tag{8}$$

where $P(S)$ is the probability that the subset of nodes receiving a packet from node i is S:

$$P(S) = \prod_{j \in C(i)} (p_{ij} \mathbf{1}_{j \in S} + (1 - p_{ij}) \mathbf{1}_{j \notin S}).$$

By plugging the above expression of R_{iJ}^{any} into equation (7) we obtain a different instance of LCOR that computes the least-cost routes under the use of ERS-any. Note that not only the *costs* of routes will be different with ERS-any than ERS-best; the opportunistic routes themselves will in the general case be different, because the J minimizing (7) may not be the same under different expressions of R_{iJ}. One way to see this is that with ERS-any, a neighbor with a high D_k that is added to the CRS is more likely to be used than with ERS-best, and so the optimal CRS with ERS-any tends to be smaller than with ERS-best.

4.2 Duplicate relays

An important challenge in OR is the design of a coordination protocol to implement an ERS policy. This protocol must ensure that the nodes receiving a packet agree on their identities, and select the correct relay as required by the ERS policy. While an ideal protocol executes the ERS policy with complete reliability, it is in practice possible that the outcome of executing the coordination protocol is incorrect. One such error would be that more than one receiver forwards a packet. Such a duplicate transmission could happen, for example, when due to lost signalling information, two nodes mistakenly believe they are each the only receiver of a packet.

In addition to accounting for different ERS policies, the LCOR framework and algorithms can also capture imperfect (e.g, real) coordination protocols that do not always carry out the ERS decision correctly. For example, consider an implementation of ERS-any where each node other than the effective relay mistakenly forwards a duplicate packet is forwarded with probability p. In such a case, the RPC can be expressed as:

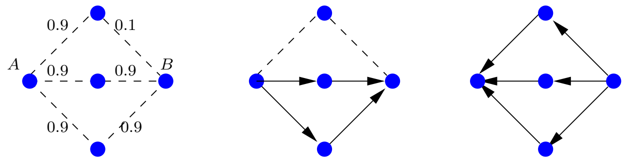
$$R_{iJ}^{dup} = (1 + p(|J| - 1)) \cdot R_{iJ}^{any}. \tag{9}$$

This R^{dup} can then be used as the RPC in the LCOR algorithm which then takes into account the expected cost of duplicates. The result is that sizes of CRS in an LCOR route is smaller when using as RPC R^{dup} than R^{any}, because the possibility of duplicates increases the cost of having large CRS's, which can partially (or entirely) offset the reduced forwarding cost captured by the ALC. This effect of the LCOR algorithm "clamping down" on CRS sizes is actually dependent on the distance to the destination. At close distance (e.g, 1 or 2 hops away), the overall penalty of transmitting a duplicate is less steep than at far distances, since the duplicate will be redundantly transmitted over a smaller number of hops. A comparison of LCOR routes found by the algorithm using R^{dup} vs. R^{any} is shown in Figure 6, with a high value of $p = 0.1$ in order to make the distance-dependent CRS size reductions clearly apparent.

4.3 Asymmetry

With single-path routes, route costs are symmetric as long as individual links are symmetric. This property does not hold for opp. routes. Figure 7(a) shows a network with two end-points A and B, and three intermediate nodes. All links have delivery probability 0.9, except for one link that has delivery probability 0.1. The ALC metric is expected transmission count (ETX).

Figure 7(b) shows the least-cost opp. route from A to B. This route does not use the upper node as a candidate relay, because it has a poor connection to B. Given that this upper node has to re-transmit on



(a) Network topology (b) LCOR route from A to B. (c) LCOR route from B to A.

Figure 7: Example of a least-cost opp. route that is not symmetric. Link delivery probabilities are depicted in the left-most figure. The cost metric is expected transmission count (ETX).

average 10 times to deliver a packet to B, it is preferable for node A to re-transmit in the rare case that neither of the two bottom candidates receives the packet, even if the upper node has received it.

Now let us consider the reverse direction, from B to A. Here, the least-cost opp. route uses all three intermediate nodes are candidate relays. Using a smaller CRS set would result in a higher ETX to get from B to the set, and since all intermediate nodes have the same delivery probability to A, there is no performance hit from using the upper relay (unlike when sending from A to B).

5. RELATION TO EXISTING WORK

Link-layer anycasting has been previously proposed and motivated in various forms [1] [6]. These works focus on *mechanisms* to implement anycast forwarding at the link layer, and assume that the network layer maintains a list of possible relay candidates (e.g., by a multi-path routing protocol) that is provided to the link layer. These works do not propose specific strategies for the selection of these candidates by the routing protocol, and the LCOR algorithm could be used to feed these link layers with relay candidates.

Jain and Das [4] go a step further by integrating an anycast extension of the 802.11 link layer with the multi-path AODV (AOMDV) [9] routing protocol. They observe the same tradeoff as [6] between number of candidates and path length. Motivated by an empirical evaluation, they modify AOMDV to allow the use of paths up to *one* hop longer than the shortest path.

Note that the original design goal of most multipath routing protocols is usually to improve load-balancing, redundancy or failover by providing multiple route choices. This is in contrast with LCOR (and OR in general) that provides multiple relay candidates specifically to take advantage of anycast forwarding. In the context of wired networks, one example of multipath routing is the work of Zaumen and Garcia-Luna-Aceves [10]. This work defines a routing algorithm that computes the multipaths containing all paths from the source to the destination that are guaranteed to be loop-free at every instant. The definition of opportunistic route in Section 3 is similar to theirs, but our notion of *least-cost* opportunistic routes is different, because our cost model is designed to reflect the use of anycast forwarding.

One approach to candidate selection is to use geographic positions [11], and select as candidate relays those nodes that are closer to the destination than the current node. This approach is simple and trivially guarantees loop-freedom. One challenge inherent in a geographic approach is however that radio propagation is highly irregular at local scales, and so making progress in physical distance does not guarantee making progress in the actual network topology.

Finally, Zhong et. al. previously remarked [5] that the routes used by ExOR are not optimal, and propose a heuristic-based method for candidate relay selection based on single-path metrics.

6. CONCLUSION

This paper introduces an algorithm to compute least-cost opportunistic routes in multi-hop wireless networks. The technique is general and the associated framework can accommodate a number of different network and cost models.

7. REFERENCES

[1] P. Larsson, "Selection Diversity Forwarding in a Multihop Packet Radio Network with Fading Channel and Capture," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, pp. 47–54, 2001.

[2] B. Zhao and M. C. Valenti, "Practical Relay Networks: A Generalization Of Hybrid-ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 23, 2005.

[3] S. Biswas and R. Morris, "Opportunistic Routing in Multi-Hop Wireless Networks," in *Proceedings of the ACM Symposium on Communications Architectures and Protocols (SIGCOMM)*, Philadelphia, USA, 2005.

[4] S. Jain and S. R. Das, "Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks," in *WoWMoM'05*, 2005.

[5] Z. Zhong, J. Wang, G.-H. Lu, and S. Nelakuditi, "On Selection of Candidates for Opportunistic AnyPath Forwarding *(Extended abstract)*," in *Poster Session of ACM MOBICOM'05*, August 2005.

[6] R. R. Choudhury and N. H. Vaidya, "MAC-layer Anycasting in Ad Hoc Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 75–80, 2004.

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

[8] H. Dubois-Ferrière, *Anypath Routing*. PhD Thesis #3636, EPFL, 2006.

[9] M. Marina and S. Das, "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks (AOMDV)," 2001. [Online]. Available: citeseer.ist.psu.edu/marina01demand.html

[10] W. T. Zaumen and J. J. Garcia-Luna-Aceves, "Loop-Free Multipath Routing Using Generalized Diffusing Computations," in *INFOCOM*, 1998, pp. 1408–1417.

[11] R. R. M. Zorzi, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance," *IEEE Trans. on Mobile Computing*, vol. 2, 2003.