Least-Cost Opportunistic Routing

ABSTRACT

In opportunistic routing, each node maintains a group of candidate next-hops to reach a particular destination, and transmits packets to *any* node in this group. The choice of candidate next-hops at each node is a key question that is central to the performance of opportunistic routing.

This paper addresses the *least-cost opportunistic routing* (LCOR) problem: how to assign the set of candidate next-hops 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 next-hops 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.

The LCOR algorithm and framework are general and can be applied to a variety of networks and cost models, including and beyond ETX-based metrics to improve throughput with lossy links. This paper further focuses on the application of LCOR to low-power wireless communication, and introduces a new link-layer technique to decrease energy transmission costs in conjunction with opportunistic forwarding. The design is implemented and evaluated on a 50-node wireless testbed. Simulation and testbed results demonstrate reductions in energy transmission by up to a factor of three compared to standard routing, up to 30% compared to opportunistic routing using single-path metrics. Furthermore LCOR routes are more robust and stable than with approaches based on single-path distances, due to the integrative nature of the LCOR's route cost metric.

1. INTRODUCTION

In many wireless networks, it is less costly to transmit a packet to *any* node in a set of neighbors than to one specific neighbor. 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 routing 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 a priori, at each node, which neighbors should be candidate next-hops to reach a destination, and how to prioritize and select the effective next-hop when multiple candidates have received a transmission.

Previous work has focused on mechanisms for link-layer anycasting, and on the difficult challenge of devising a robust, low-overhead coordination protocol for receivers of a packet to agree upon a next-hop [1–6] when multiple candidates receive a packet. The seminal work of Biswas and Morris [3] additionally introduces a whole-system design and implementation on a live 802.11 network, with demonstrable performance benefits over single-path routing. At the same time, comparatively little attention has been given to the problem of how to best select and prioritize candidate next-hops so as to minimize routing costs.

This paper builds upon previous work in opportunistic routing and extends it by revisiting the question of candidate next-hop selection. The starting point of this work is the question: with OR, are there practical and general ways to compute the *optimal* (under a given cost and network model) candidate next-hops to be used at each node to reach a given destination?

The optimal selection of candidate next-hops must take into account conflicting tradeoffs. On the one hand, employing more than one candidate next-hop usually decreases 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 our routing topology. Also, as the number of candidate next-hops grows, so does the overhead of link-layer coordination and the risk of costly duplicate transmissions.

The challenge lies essentially in the fact that with OR, each packet can traverse a multitude of possible paths to reach a destination, with each path possibly having a different cost. Which path each packet follows 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. As such, each possible choice of candidate next-hops 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. An essential aspect to the practicality of our solution is that it does not require explicitly computing this distribution.

The solution to the problem of finding optimal candidate next-hops for OR lies in 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. As such, the cost of anycasting to a set (which is often different than the cost of unicasting to a single node) is explicitly considered and is central to computation of routes. A node transmits a packet to *any* node in this set using link-layer anycast. 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 next-hops. Within this framework, we formulate a distributed algorithm *for least-cost opportunistic routing (LCOR)*. The LCOR algorithm is operationally similar to the classical distributed Bellman-Ford, but is driven by different metrics that generalize unicast link and path costs respectively. This algorithm provably computes the optimal choices of candidate next-hops. Note however that any routing protocol is only 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). As is the case for other routing protocols, the notion of optimality is here relative to the model of a network.

Prior work on opportunistic routing has primarily focused on improving throughput. This paper also introduces a new link layer technique for low-power, low-rate wireless networks called anycast low-power listening (A-LPL), which allows to decrease energy consumption of wireless interfaces by exploiting opportunistic routing. The scheme is implemented and evaluated in a 50-node wireless testbed.

In summary, this paper makes three main contributions. The first is a theoretical framework that encompasses a wide spectrum of opportunistic protocols with a common set of concepts and metrics. The second is LCOR, a distributed algorithm that finds the optimal assignment of candidate next-hops at each node for a given destination. The third is the application of LCOR to low-power wireless networks, and a performance evaluation combining simulation and live measurements over a 50-node testbed. We believe that this work broadens the scope and relevance of opportunistic routing, and provides a useful framework for the further evaluation, analysis, and design of new OR protocols.

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 shows an application of LCOR to low-power wireless networking, and Section 6 evaluates performance. Section 7 describes related work, and Section 8 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 next-hop set** (CNS), denoted $J(i)$ (or J); it contains all the nodes which may be used as next-hops for packets forwarded by i toward the destination[1].

Notation and Acronyms

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

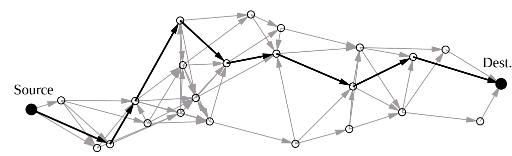


Figure 1: An opportunistic route is the union of all possible paths from a source to a destination, induced by the choice of candidate next-hops at each node. A possible trajectory through the opp. route is highlighted in bold.

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 CNS 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 CNS $J(n_1), J(n_2), \ldots J(n_k)$ 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. One example of unicast link cost for wireless networks is the expected transmission count (ETX) [7] metric, which counts the expected number of transmissions to successfully deliver a packet across an unreliable link. If each packet transmitted by node i to j is independently received with probability[2] p_{ij}, the ETX from i to j is $d_{ij}^{ETX} = \frac{1}{p_{ij}}$.

With OR, we must generalize the notion of link cost from single-path routing, to account for anycast for-

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

[2]This definition considers that ACKs are delivered reliably; accounting for the possibility of ACK loss simply requires modifying eq. (2) to replace each p_{ij} by the probability of successful packet reception *and* ACK delivery.

warding 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, we can use an ETX-like metric, and generalize the unicast ETX to the expected number of transmissions until any node in J receives the packet. Its expression is:

$$d_{iJ}^{ETX} = \frac{1}{p_{iJ}}, \tag{1}$$

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

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

Note that this metric generalizes the unicast ETX, that is, for a singleton CNS with $|J| = 1$, the anycast ETX reduces to the unicast expression.

Similarly to unicast link costs, the choice of anycast link cost is a modelling decision, and should reflect the cost criterion that we wish to minimize. Other anycast link costs are possible, and another example is given in Section 5.

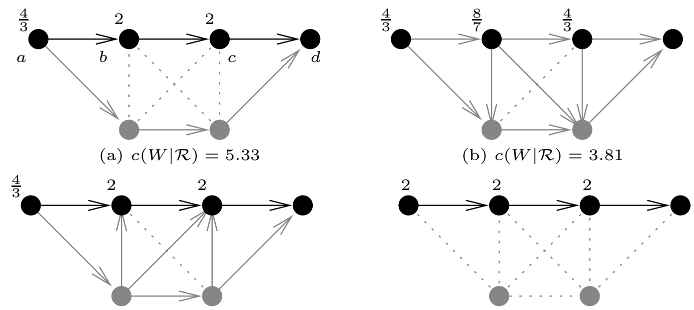
2.1.2 Cost of a trajectory in an anypath route

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 nodes in T:*

$$c(T|\mathcal{R}) = \sum_{i \in T} d_{iJ(i)} = d_{sJ(s)} + d_{n_1 J(n_1)} + d_{n_2 J(n_2)} + \ldots d_{n_k J(n_k)}.$$

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 next-hop set J, and not just on the effective next-hop 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 next-hops, and so its ALC is $d_{aJ(a)}^{ETX} = (1 - 0.5^2)^{-1} = 4/3$. Nodes b and c have a single candidate next-hop 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 next-hops. 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.



(a) $c(W|\mathcal{R}) = 5.33$ (b) $c(W|\mathcal{R}) = 3.81$ (c) $c(W|\mathcal{R}) = 5.33$ (d) $c(W|\mathcal{R}) = 6$

Figure 2: Cost of the same trajectory $T = (a, b, c, d)$ traversing four different opportunistic routes. The cost $d_{iJ(i)}^{ETX}$ 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 $P(T)$. 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,*

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

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

This opportunistic route cost generalizes the cost of a single path route: if all CNS'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.

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.*

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. Note also that 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.

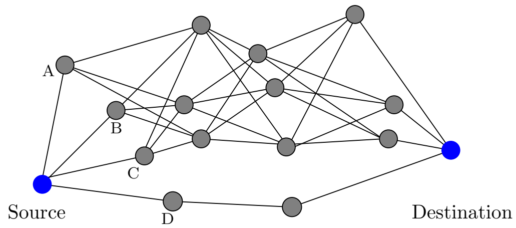


Figure 3: Mis-match of single-path metrics with opportunistic routing. Sending a packet via the dense mesh takes advantage of anycast forwarding and is often cheaper than via the four-node strand at bottom, even if it goes through more hops. However, the use of a single-path metric prevents the source 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.

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 next-hops 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 next-hops all neighbors with lower single-path cost to reach the destination. Before developing our solution to the LCOR problem, we discuss why strategies based on shortest-path metrics do not always lead to optimal CNS 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 next-hops 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 next-hop 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 next-hops 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 next-hops 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 in line with intuition, 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. Indeed, the direct way of computing the expectation would be to enumerate all possible trajectories and compute the probability and cost of each, which is hardly feasible in a distributed, wireless setting.

3.1 Remaining path cost

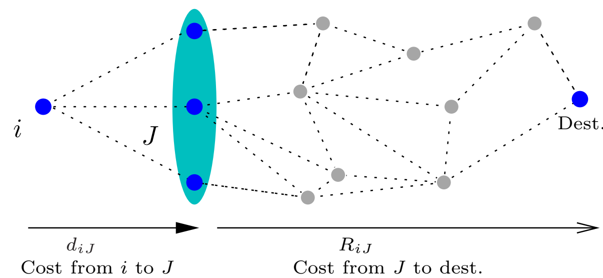


Figure 4: The cost of an opportunistic route can be separated into two components: the anycast link cost, which is the cost to reach the next hop, and the remaining path cost, which is the cost to get from the next hop to the destination.

With unicast forwarding, it is trivial that the remaining cost for a packet to reach the destination after it is forwarded to the next-hop is the path cost from the next-hop to the destination. With anycast forwarding, the effective next-hop 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 CNS 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 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 be at first 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 ends up effectively being the next hop and forwarding 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 becomes the next hop. If multiple nodes in J receive the packet, then the receiver with lowest cost to reach the destination is selected as the next hop. 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 $R_{iJ} = D$. If all nodes in J receive all packets from i, then $R_{iJ} = \min_{k \in J} D_k$. 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

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

where it is assumed (without loss of generality) that the nodes in J are sorted by their distance 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:

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

Note that like the anycast link cost, the RPC generalizes the single-path case: when $|J| = 1$, it simply becomes the cost from the next-hop to the destination. Note also that the same CNS J can give a different RPC for two different senders i, since this RPC is affected by the delivery probabilities from the sender to each candidate next-hop. In other words, the remaining path cost from a CNS J to the destination depends not only on J itself, but also on the predecessor node i of J.

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, there is nonetheless one technical criterion which they must satisfy together in order for the routing algorithm to operate correctly. This is the physical cost criterion; it is met by all costs used in this paper.

Definition 4. *Consider a node i with CNS 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.

Less formally, the physical cost criterion says that if a node i adds to its CNS a neighbor with higher cost to the destination than i itself, then i's cost to reach the destination will increase. This can be seen as analogous to the requirement that link costs be non-negative in order for single-path routing to converge.

3.3 Least-cost opportunistic routing algorithm

How does a node select which of its neighbors should be candidate next-hop 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 $J \subseteq N(i)$:

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

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:

$$D_1^h = 0, \qquad \text{for all } h, \tag{6}$$

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:

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

where R_{iJ}^h is the remaining path cost computed using the costs D_j^h, $j \in J$ from the previous iteration. The CNS 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 the Appendix. The LCOR algorithm resembles the classical Bellman-Ford algorithm, with the crucial difference that the cost metrics are generalized to handle next-hop candidate sets rather than next-hop 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.

The upper bound on the algorithm's convergence time (in number of iterations) is the same as for single-path Bellman-Ford. The complexity of the LCOR algorithm is however greater, since there are $2^{|N(i)|}$ possible subsets that must (in the worst case) be evaluated, compared to $|N(i)|$ possible next-hops with single-path routing. Note that given the physical cost criterion of Section 3.2, a node i needs in practice only to consider those neighbors which have advertised a cost lower than i's current estimate of D_i. Further optimizations are possible in certain cases. For example, in the LCOR instance of Section 5, the exponential search is entirely avoided, and only $|N(i)|$ possible CNS's must be evaluated. This happens because the ALC d_{iJ}^{A-LPL} in Section 5 depends only on $|J(i)|$ and not on the individual nodes in J. It is thus possible in that case to minimize (7) by sorting nodes in order of increasing cost, and evaluating the $|N(i)|$ sets of size $1, 2, \ldots |N(i)|$. It remains nonetheless that in certain practical instances the complexity of minimizing (7) may be too high; an area of future work concerns the investigation of reduced-complexity heuristics to find an approximate solution to the minimization for such cases.

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 next-hop selection

When a packet transmitted by a node i is received by more than one node in i's CNS, a decision must be made as to which receiver should then forward the packet further. We call this an effective next-hop selection (ENS) 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 *ENS-best*. The framework and algorithm outlined in the previous sections allow to model and capture the effect of other next-hop selection policies. Another example of ENS policy is *ENS-any*, where the next-hop is chosen uniformly at random among receivers of a packet.

In comparison with ENS-best, ENS-any has the disadvantage that it may select as next-hop a receiver with a more costly path to the destination than the least-cost receiver. At the same time, executing ENS-any in a protocol may have lower cost than executing ENS-best. Also, using ENS-any spreads the forwarding load more evenly than ENS-best over the entire opportunistic route. These arguments are qualitative. We do not seek to claim that ENS-any should be used over ENS-best, but rather to point out that other policies exist, and show how they can be modelled within LCOR.

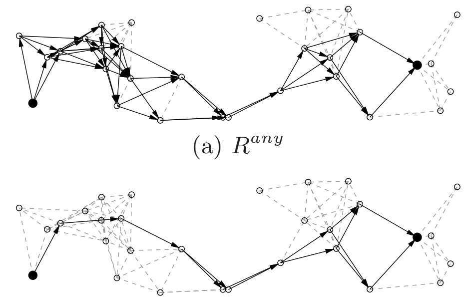
With ENS-any, if $S \subseteq J$ is the set of nodes that receives a transmission, then the remaining path cost is the average cost over the nodes in S. The remaining path cost R_{iJ}^{any} can thus be written as

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

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 ENS-any. Note that not only the *costs* of routes will be different with ENS-any than ENS-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 ENS-any, a neighbor with a high D_k that is added to the CNS is more likely to be used than with ENS-best, and so the optimal CNS with ENS-any tends to be smaller than with ENS-best.



(a) R^{any}

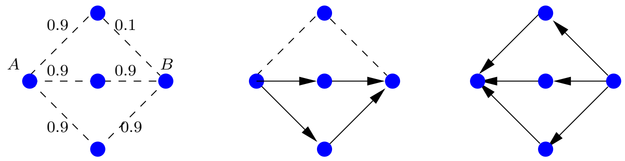
(b) R^{dup}

Figure 5: Comparison of least-cost opportunistic routes with perfect link layer coordination vs a link layer that sometimes lets through duplicate transmissions.

4.2 Duplicate next-hops

An important challenge in OR is the design of a coordination protocol to implement an ENS policy. This protocol must ensure that the nodes receiving a packet agree on their identities, and select the correct next-hop as required by the ENS policy. While an ideal protocol executes the ENS 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 ENS policies, the LCOR framework and algorithms can also capture imperfect (e.g, real) coordination protocols that do not always carry out the ENS decision correctly. For example, consider an implementation of ENS-any where each node other than the effective next-hop mistakenly forwards a duplicate packet is forwarded with probability



(a) Network topology

(b) LCOR route from A to B.

(c) LCOR route from B to A.

Figure 6: 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).

p^3. In such a case, the RPC can be expressed as:

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

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 CNS 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 CNS's, which can partially (or entirely) offset the reduced forwarding cost captured by the ALC. This effect of the LCOR algorithm "clamping down" on CNS 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 5, with a high value of $p = 0.1$ in order to make the distance-dependent CNS size reductions clearly apparent. We shall see that the A-LPL anycast link layer introduced in Section 5 fortunately has a very small chance of duplicates.

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 6(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 6(b) shows the least-cost opp. route from A to B. This route does not use the upper node as a candidate next-hop, because it has a poor connection to B. Given that this upper node has to re-transmit on 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 next-hops. Using a smaller CNS 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 next-hop (unlike when sending from A to B).

[3] Of course, in a real system, it is difficult to know p a priori. This probability might be estimated by nodes locally observing the outcome of coordination actions.

5. APPLICATION TO LOW-POWER WIRELESS NETWORKS

This section shows how LCOR can be applied to reduce energy consumption of packet forwarding in low-rate, duty-cycled wireless networks. Since the radio is the dominant energy consumer in many low-power wireless devices [8] [9], it is necessary to power it down whenever possible, by using some form of a duty cycling link layer. Duty-cycling schemes trade off latency for energy efficiency, and a key difficulty to achieve low duty cycles (e.g., radio utilization below 10^{-2}) is to reliably rendezvous between a sender and a receiver whose radios are turned off most of the time.

Several strategies for low-power operation of wireless links have been proposed. We focus on low-power listening (LPL) [8], a simple technique for link-layer duty cycling, and introduce *anycast LPL* (A-LPL), a derived duty cycling technique that exploits anycast forwarding to reduce energy costs in conjunction with LCOR. Note that LCOR can also be used with other low-power link schemes; we illustrate it with LPL because of the simplicity and robustness of LPL, that are evidenced by its widespread adoption in a large number of wireless sensing projects and by the number of further studies and improvements that have been made to LPL.

5.1 Low-power listening

Each node awakens once within an interval of duration t_{rx} and briefly samples the channel. If the node hears no activity on the channel, it sleeps until its next wakeup time, or until it has a packet to transmit, whichever comes first. If the node does hear activity on a channel, and specifically if the node recognizes a preamble sequence, it remains awake until it receives the packet that is sent following the preamble.

LPL is asynchronous and nodes do not keep track of their neighbors' duty cycles. Since a sender cannot simply start transmitting at the time when the destination wakes up, it precedes the packet transmission by a long *preamble* (a well-known, periodic bit sequence). In order to guarantee that the preamble will be heard by the receiver, it must last at least as long as the interval t_{rx} between node wakeups. This means that as the duty cycle is brought down (by increasing t_{rx} such that the overall fraction of time spent listening is decreased), the cost of sending a packet grows due to the increasingly long preamble. This can be viewed as the drawback to LPL's simplicity and robustness. Many optimizations to LPL are possible, such as embedding destination or offset information in the preamble. While we do not cover them here, these optimizations are compatible with the A-LPL scheme described next.

5.2 Anycast LPL

The design of A-LPL follows from the idea that if a node transmits a packet to any node in a group of neighbors which each listen at randomly distributed times, then it should be possible to reduce the length of the preamble that would be necessary when sending to one specific neighbor. The net effect is a reduction in energy cost and latency to transmit a packet.

For clarity, we assume in the remainder of this Section that links are reliable ($p_{ij} = 1$), and so do not model the ETX component in the two link costs below. Note however that the use of this scheme is complementary to the use of anycast forwarding to reduce the ETX with lossy links; both can be done in combination.

Assume that wakeup times are uniformly distributed within the interval t_{rx}. Assume that we use a preamble of length $t_{pre} < t_{rx}$. Define $\lambda = \frac{t_{pre}}{t_{rx}}$, and note that with standard (unicast) LPL we have $\lambda \geq 1$. We say that a transmission *hits* a node if the preamble covers the node's wakeup interval. The probability p_{hit} of hitting *one* specific neighbor is λ, but the probability of hitting *any* node in a CNS of size n is

$$p_{hit} = 1 - (1 - \lambda)^n. \quad (10)$$

While p_{hit} increases with the size of the CNS, guaranteeing that some node in the CNS receives a packet (e.g., reaching $p_{hit} = 1$) still requires having a preamble of length at least t_{rx}. The way to exploit this increased probability is therefore to combine the shortened preambles with a re-transmission strategy. The average number of transmissions until we hit at least one node will be $1/p_{hit}$.

An anycast link cost metric for A-LPL must reflect an entire transmission cost, which includes both preamble and packet transmission time[4]. Note the tradeoff between decreasing λ (cost of a single transmission) and increasing $1/p_{hit}$ (expected number of transmissions). The optimal point in this tradeoff depends on the size $|J|$ of the CNS and the relative durations of t_{pkt} and t_{rx}. We can now define the *energy* anycast link cost for A-LPL:

$$d_{iJ}^{A-LPL} = \min_{t_{pre} \in [0, t_{rx}]} \frac{t_{pre} + t_{pkt}}{p_{hit}} = \min_{\lambda \in [0,1]} \frac{\lambda t_{rx} + t_{pkt}}{1 - (1 - \lambda)^{|J|}}, \quad (11)$$

where the numerator is the energy cost of *one* transmission, and is multiplied by the expected number of transmissions. Note that this metric generalizes the unicast LPL cost, that is, for $|J| = 1$, we have $d_{iJ}^{A-LPL} = t_{rx} + \lambda t_{pkt}$ which is equal to the forwarding cost in the unicast case. Computing d_{iJ}^{A-LPL} analytically is hard, because minimizing eq. (11) requires finding the zeroes of an order-$|J|$ polynomial. We therefore compute it numerically, and plot it in Figure 7. The optimal tradeoff point is for small values of λ (except when $|J| = 1$), showing that with unicast transmission there is no advantage to the strategy of reducing preambles and retransmitting until a preamble hit. Using the optimal values for λ, the transmission cost is reduced by a factor of 2-5 for practical CNS sizes ($|J| \leq 10$).

[4] Energy is proportional to transmission time under the assumption of fixed transmit power.

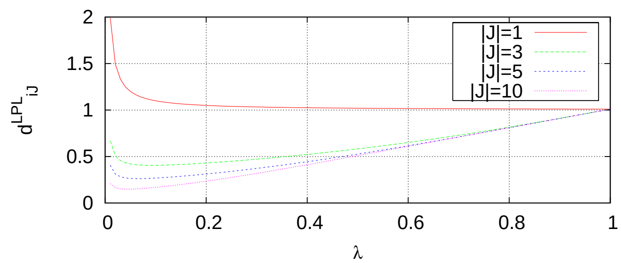


Figure 7: Anycast link cost d_{iJ}^{A-LPL} as a function of preamble length λ for different CNS sizes. In this figure we have set $t_{pkt} = 0.01 \cdot t_{rx}$, in line with existing implementations of LPL [8] [9]. With a CNS of size 3, transmission cost is reduced by a factor of 2.5 over unicast LPL, with a CNS of size 10 it is reduced by a factor greater than 5.

To compute the remaining path cost with this anycast forwarding mechanism, note that at each (re-)transmission, the probability of any node in $J(i)$ receiving the packet is the value p_{hit} obtained in (10). Thus, the remaining path cost is obtained by substituting λ_{opt} for p in (4), where λ_{opt} is the argument minimizing (11).

5.3 Link-layer coordination with A-LPL

As discussed in Section 4, having an efficient and robust link-layer coordination protocol is a critical task when using OR with ETX-based networks [3]. With A-LPL however, the burden on a candidate coordination protocol is much smaller, because it is very rare that multiple nodes receive a same packet, due to their radios being turned on only a very small fraction of the time (and under our independence assumption). Figure 7 shows that the optimal value of the preamble length λ is small. Thus, at each packet retransmission, the probability that multiple nodes receive the packet is low. In most cases, a preamble hit happens for a single node at a time, and there is no need for a costly coordination phase between multiple nodes. In fact, in the implementation of the following section, each node in a CNS that receives a packet forwards it; a coordination protocol would not be worth its cost given the minute number of duplicates that actually happen.

6. PERFORMANCE

This section evaluates the performance of LCOR in comparison with standard single-path routing and with opportunistic routing using single-path metrics. This evaluation is achieved by means of simulations under an intentionally simple network and channel model, and with measurements from a full implementation of LCOR running on a 50-node testbed, that subjects the routing algorithm to the vagaries of real-world wireless channel characteristics. This evaluation focuses on low-power routing with LPL and do not cover throughput performance under ETX due to lack of space. We note however that that the properties and results shown here hold at least qualitatively under the ETX metrics (2), (4).

In this section we use the following terminology: *SP* routes are shortest single-path routes as found by classical Bellman-Ford or Dijkstra algorithms. *LCOR* routes are the least-cost opp. routes found by the algorithm of Section 3. Finally *SP-OR* routes are opportunistic routes obtained using a single-path metric (such as in ExOR) as discussed in Section 2.2.

6.1 Route costs

We first evaluate the cost of paths found by LCOR. We simulated a network with nodes uniformly distributed in a square surface. Connectivity is determined exclusively by distance, e.g. we use the unit disk graph model. All simulations reported here use average node density of 10 and networks with 500 nodes. For graphs that plot an empirical mean as a function of some underlying variable (i.e., graphs that do not plot an empirical cumulative density function), we run simulations until the 95% confidence interval is less than ±10% of the empirical mean.

The first set of simulations evaluates the cost of SP, SP-OR, and LCOR routes. For each node pair, we compute the shortest single-path route and the LCOR route. We then order nodes by single-path distance, and in Figure 8(a) plot the average LCOR costs as a function of this shortest single-path distance. The LCOR route costs are reduced by a factor between 1.8 and 2 compared to SP costs. Furthermore, the gap widens for diminishing duty-cycle ρ, due to the relative cost of a retransmission becoming smaller as t_{pkt} decreases relative to t_{rx}. Therefore, the minimization in (11) can use lower values of λ as t_{rx} is increased. Note that LCOR route costs are (roughly) a constant factor of single-path costs; the cost reduction of LCOR routes increases with *density* rather than *diameter*.

Figure 8(a) compares LCOR route costs with those of SP-OR routes. The cost of the SP-OR routes is approximately 40% higher than that of LCOR routes.

One way to characterize an opportunistic route is to consider the number of candidate next-hops that nodes in this route have. Do LCOR routes have more candidate next-hops than SP-OR routes? Or is the lower cost of LCOR due to the choice of CNS being informed by the more suitable ALC metric, but with similar CNS sizes? To answer this question, we define the *average out-degree* as the empirical average of $|J(i)|$ for nodes at a given shortest-path distance to the destination, and plot it in Figure 8(b), averaged over 10000 network realizations. This shows us that, at least under the d_{iJ}^{A-LPL} cost model, LCOR routes are able to use more candidate next-hops than SP-OR routes; nodes in LCOR routes have about 4 candidates, in comparison with 2 for SP-OR routes.

6.2 Robustness

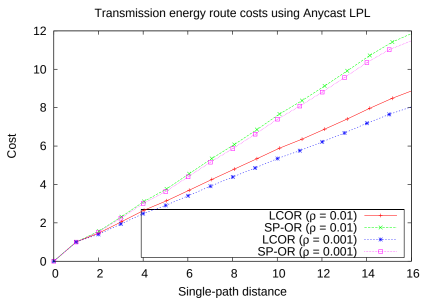
Route costs are a primary measure of a routing algorithm's performance, but are not the only measure. Robustness is another important property of any algorithm that is intended to run in a distributed wireless setting. One essential aspect of robustness is the resilience of routes in the face of topology changes. We studied this resilience by running the following simulation experiments. First we generate a network realization, and compute all least-cost opp. routes in it. Then, we randomly remove a number of links in this network. Links are independently removed with probability p_v, and we then count the number N_d of routes that are disconnected in the new topology. A single link cut is sufficient to disconnect an SP route; for LCOR or SP-OR routes disconnectedness means that there is no path to reach the destination.

We plot the empirical CDF of N_d in Fig. 8(c). As expected, LCOR routes have fewer disconnections than SP. More interesting however is that LCOR routes are also significantly less prone to disconnection than SP-OR routes. For example, the probability that *less* than 10% of routes are disconnected with LCOR is over 0.95, while it is only 0.65 with SP-OR. This is a direct consequence of LCOR's larger CNS sets as shown in Figure 8(b). Note that robustness is a multi-faceted property, and a complete investigation should also examine the cost of routes computed with an *approximate* (or noisy) view of network topology, as is often the case in wireless networks. The intuition is that the integrative nature of the LCOR cost metric provides routes that are more stable to such noisy inputs than SP or SP-OR.

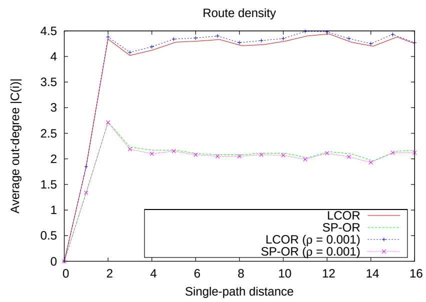
6.3 Testbed and Implementation

We implemented LCOR in conjunction with A-LPL on the *TinyNode* wireless platform, which is targeted at low-power embedded sensing applications. We refer to [9] for more details on the platform and its radio device. Our implementation runs under TinyOS 2.0, and its salient features are as follows. At the bottommost radio interface layer, packets can be sent with a varying-length LPL preamble, that is selected by the anycast link layer based upon the duty-cycle and the number $|J(i)|$ of candidates. The value of λ_{opt} obtained by minimizing (11) is stored in a lookup table containing the value of λ_{opt} for each possible value of $|J(i)|$.

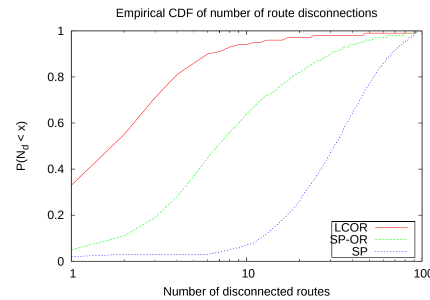
The implementation currently uses a very simple link estimation algorithm that keeps count of the (estimated) number of lost packets over a window of k previously transmitted packets from each neighboring node (with $k = 32$ in our implementation). The link estimation algorithm also assumes a minimum broadcast rate from each node, allowing it to progressively degrade a link estimate when no packets are received from a neighbor for an extended interval. The link estimator simply characterizes a link as "on" or "off"; it does not seek to estimate the *probability* of packet delivery or other derived metrics such as ETX. A link is "on" whenever it has delivered at least $k-4$ of the previous k transmitted



(a) Comparison of average LCOR costs using ALC metric d_{iJ}^{A-LPL}, as a function of shortest single-path distance, and for two different duty cycles.



(b) Average outdegree $|J(i)|$ as a function of single-path distance to the destination.



(c) Empirical CDF of N_d, the number of disconnected routes, for the link addition/deletion model with $p_v = 0.05$.

Figure 8: Simulation results for a 500-node network and average density 10.

packets[5]. If no links satisfy this criterion, it is relaxed until at least 5 neighbors are considered reachable.

Note that unlike many existing approaches, this link estimation scheme does not exchange reverse link estimates in order to compute bidirectional link quality. This simplification is motivated by the observation that link asymmetry is usually less pronounced on highly reliable links [10]; since we only attempt to use highly reliable ("on") links, we assume that asymmetry will pose less problems than if we frequently used intermediate and poor links.

Moving up to the networking layer, nodes advertise their route costs by means of periodic route beacon broadcasts. Each beacon contains the sender address, the destination to which it is advertising a route, the sender's cost to reach that destination, and a destination sequence number that serves to prevent routing cycles.

We ran our LCOR implementation on a testbed consisting of 50 TinyNodes deployed over 3 floors of a campus building. Each node is equipped with a serial-to-ethernet adaptor allowing to use the building's wired infrastructure to reliably report statistics. We instrumented the radio stack to keep records for the last 20 packet transmissions. For each outgoing packet (including both locally originated and forwarded packets), our instrumented code logged the originator address, a 16-bit sequence number placed by the originator in the packet payload, and the total number of transmitted bytes. This last value counts *every* transmitted byte, including both the preamble and the packet itself. It therefore allows us to account for and compare the costs of variable-sized preamble lengths used by nodes having different CNS sizes $|J|$.

6.4 Experimental Results

Each of the experiments described in this section used the testbed and implementation described previously, and ran over varying configurations of transmit power (5 dBm and 15 dBm) and choice of sink node (either node 14 or node 48, each one being placed at opposite sides of the building).

The combination of two transmit power settings and two sink choices gives us a total of 4 experiment configurations. Each configuration was run once using LCOR routing and once with SP routing, each time for three hours, giving a total running time of 24 hours. For single-path routing we used the same protocol, but with the network layer constrained to select candidate next-hop sets of size 1. This allows a fair comparison between LCOR and SP that is not affected by protocol implementation differences, as would have been the case if we used an entirely separate single-path protocol.

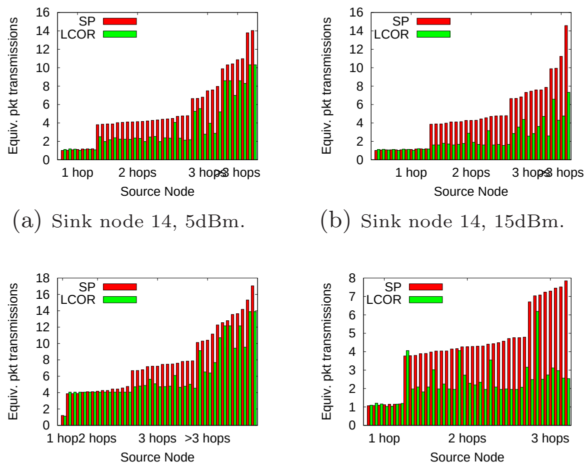
Nodes ran at a 0.8% duty cycle, giving a wakeup period $t_{rx} = 300ms$, and a listen time of $t_l = 2.12ms$. Data packets were originated at an average rate of one packet per minute at each node. Routing beacons were transmitted every 10 minutes, except for the first 5 minutes of each three hour run, where routing beacons were transmitted every 30 seconds in order for routes to converge more rapidly.

This data is represented in Figure 9, with one plot for each experiment configuration described above. Each bar in a plot represents the total equivalent packet transmissions (total number of bytes transmitted, normalized by the length of a packet without the preamble), averaged over all packets originated by a given source node. Nodes are ordered by increasing single-path cost. Average single-path hop distances are annotated along the x-axis; the "step" between the costs at successive hop-counts narrows as distance increases, due to the fact that this cost is the sum of an increasing number of random components. Unsurprisingly, overall costs are lower at 15dBm than at 5dBm, because paths become shorter with increased network density. The relative advantage of LCOR over SP is also greater at 15dBm, because larger CNS sizes allowed denser neighborhoods enable A-LPL to further reduce preamble lengths.

Most importantly, these plots show that least-cost opportunistic routing decreases costs for all nodes but one (in Figure 9(d)); for packets originated at distances greater than one hop from a sink LCOR reduces trans-

[5] These constants were chosen based on the informal observation that many "good" wireless links approach 100% packet delivery at short timescales.

missions by a factor of up to 3.



(a) Sink node 14, 5dBm. (b) Sink node 14, 15dBm.

(c) Sink node 48, 5dBm. (d) Sink node 48, 15dBm.

Figure 9: Average packet delivery costs for SP and LCOR. Each plot represents one configuration of transmit power and sink node. Nodes are ordered by increasing shortest-path cost.

While reduced transmission energy costs are a key performance indicator, they alone do not determine the performance of a wireless routing protocol. In particular, end-to-end delivery must be taken into account. Table 1 shows the packet delivery rates for both SP and LCOR, averaged over all nodes in the network.

Sink	TX Power	LCOR	SP
14	5dBm	92.1%	94.2%
14	15dB	96.1%	88.1%
48	5dBm	92.4%	90.1%
48	15dBm	94.5%	86.3%

Table 1: Packet delivery rates for single-path and LCOR routing.

7. RELATION TO EXISTING WORK

Link-layer anycasting has been previously proposed and motivated in various forms. Larsson [1] proposed a joint forwarding and link layer protocol where a data frame is multicast to a set of candidate nodes. Each receiver sends back an ACK, and the sender then issues a *forwarding order* to the chosen next-hop. Choudhury and Vaidya [6] propose a similar mechanism, the main difference being that the next-hop decision is made *before* transmission; in this case there is no need for the ACK and forwarding order of Larsson's scheme.

These works focus on *mechanisms* to implement anycast forwarding at the link layer, and assume that the network layer maintains a list of possible next-hop 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 next-hop 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) [11] 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 next-hop 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 [12]. 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 [13], and select as candidate next-hops those nodes that are closer to the destination than the current node. This approach is simple and trivially guarantees loop-freedom. However, geographic positions are currently not available to most wireless devices. Also, 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.

This paper is not the first to consider opportunistic routing in the context of low-rate wireless sensor networks. Parker and Langendoen evaluated in simulation Guesswork, a protocol similar to ExOR using existing low-power link protocols [14]. They do not modify these link protocols however to specifically take advantage of anycast forwarding.

Finally, Zhong et. al. previously remarked [5] that the routes used by ExOR are not optimal. They also propose a heuristic-based method for candidate next-hop selection based on single-path metrics. Initially all nodes closer in shortest-path distance are candidates (as in ExOR); then certain candidates are *prunes* in order to reduce coordination overhead. Their work introduces a metric similar to the remaining path cost defined in various forms in this paper, but which is specific to ETX [7] costs.

8. 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. One such example is in low-power wireless networks, where we show how least-cost opportunistic routing can benefit from the use of a novel low-power anycast link-layer. The algorithm is practical and has been implemented on embedded wireless nodes.

APPENDIX: Proof of Proposition 1

We must first prove an auxiliary but important proposition.

PROPOSITION 2. *Let $\mathcal{R}$ be a LCOR route from a source to a destination, and node k be an interior node in $\mathcal{R}$. Call $\mathcal{R}_k$ the opportunistic sub-route of $\mathcal{R}$ from node k, and define $D_k = C(\mathcal{R}_k)$. Then, $D_k = D_k^*$, where D_k^* is the cost of the least-cost opp. route from k to the destination.*

PROOF. Call $\mathcal{T}$ the least-cost opportunistic route from node k to the destination. We have therefore $C(\mathcal{T}) = D_k^*$. Since $\mathcal{T}$ is the least-cost opportunistic route from k to the destination, we cannot have $D_k^* > D_k$, or otherwise $\mathcal{R}_k$ would be a shorter opportunistic than $\mathcal{T}$. It now remains to be shown that we cannot have $D_k^* < D_k$. We now proceed by contradiction and assume that $D_k^* < D_k$.

Return now to the least-cost opportunistic route $\mathcal{R}$ that $\mathcal{R}_k$ is an opportunistic sub-route of. If $D_k^* < D_k$, then any packets arriving at k from the source of route $\mathcal{R}$ toward the destination can be forwarded using $\mathcal{T}$. This results in a new route that we call $\mathcal{R}^*$, going between the same source and the destination as route $\mathcal{R}$. To complete the proof we observe that $\mathcal{R}^*$ has lower cost than $\mathcal{R}$, contradicting our initial assumption that $\mathcal{R}$ was a LCOR route. □

We now return to the proof of Proposition 1. We prove the first part of the proposition by induction over h.

Case $\mathbf{h = 1}$. Using (7) and our initial conditions, we have for all $i \neq 1$ that $D_i^1 = d_{i1}$ which is indeed the cost of the least-cost (≤ 1) opp. route to the destination.

Induction over h. We assume that D_i^h is equal to the cost of the least-cost ($\leq h$) opportunistic route from i to 1, and must show that D_i^{h+1} is equal to the cost of the least-cost ($\leq h+1$) opportunistic route. There are two possible cases for each node i. The first is that the least-cost ($\leq h+1$) opportunistic route from i to 1 contains a longest path with h or less hops. We call this route $\mathcal{R}_i^h$, and in this case we have $C(\mathcal{R}_i^h) = D_i^h$. The second possible case is that the least-cost ($\leq h+1$) opportunistic route from i to 1 contains a longest path with $h+1$ hops. Call this route $\mathcal{R}_i^{h+1}$. It has cost

$$C(\mathcal{R}_i^{h+1}) = d_{iJ(i)} + R_{iJ(i)}$$

This route consists of $|J(i)|$ links from i to each node in its CNS $J(i)$, and then of $|J(i)|$ opportunistic sub-routes from each node in $J(i)$ to 1 that each have a h-hop longest path. From Proposition 2, we know that these sub-routes must be least-cost opportunistic routes. Given this structure, there is no possible CNS among i's neighbors that has a lower cost to reach the destination with ($\leq h$) paths:

$$C(\mathcal{R}_i^{h+1}) = d_{iJ(i)} + R_{iJ(i)} = \min_{J \in 2^{N(i)}} [d_{iJ} + R_{iJ}^h] = D_i^{h+1}.$$

Calling S_i^{h+1} the least-cost ($\leq h+1$) opportunistic route cost from i to 1, these two cases thus give:

$$\begin{aligned} S_i^{h+1} &= \min\left\{C(\mathcal{R}_i^h), C(\mathcal{R}_i^{h+1})\right\} \\ &= \min\left\{D_i^h, \min_{J \in 2^{N(i)}} [d_{iJ} + R_{iJ}^h]\right\} \\ &= \min\left\{D_i^h, D_i^{h+1}\right\} = D_i^{h+1}, \end{aligned}$$

and so D_i^{h+1} is the cost of the least-cost opp. route from i to 1. The second part of the proposition follows simply from the first part and the fact that in a network with $|\mathcal{N}|$ nodes, the longest possible path has at most $|\mathcal{N}| - 1$ hops.

9. 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] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor networks," in *Proceedings of ACM Sensys*, Los Angeles, USA, April 2003.

[9] H. Dubois-Ferrière, R. Meier, L. Fabre, and P. Metrailler, "Tinynode: A Comprehensive Platform for Wireless Sensor network Applications," in *Fifth International Conference on Information Processing in Sensor Networks (IPSN)*, Nashville, Tenessee, April 2006.

[10] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," in *Proceedings of ACM Sensys*, Los Angeles, USA, April 2003.

[11] 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

[12] W. T. Zaumen and J. J. Garcia-Luna-Aceves, "Loop-Free Multipath Routing Using Generalized Diffusing Computations," in *INFOCOM (3)*, 1998, pp. 1408–1417. [Online]. Available: citeseer.ist.psu.edu/zaumen98loopfree.html

[13] 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.

[14] T. Parker and K. Langendoen, "Guesswork: Robust Routing in an Uncertain World," in *2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Nov. 2005.