Abstract—In many networks, it is less costly to transmit a packet to any node in a set of neighbors than to one specific neighbor. A well-known instance is with unreliable links, where the probability that at least one node out of \(n\) receives a packet increases with \(n\). This observation was previously exploited, by modifying single-path routing to assign to each node group of candidate next-hops for a particular destination.

This paper introduces a general formulation of the shortest anypath routing problem: how to assign a set of candidate relays at each node for a given destination, such that the cost of forwarding a packet to the destination is minimized. The key is the following tradeoff: on the one hand, increasing the number of candidate relays decreases the forwarding cost, but on the other, this increases the likelihood of “veering” away from the shortest-path route.

I. INTRODUCTION

In many networks, it is less costly to transmit a packet to any node in a set of neighbors than to one specific neighbor. For example, in a wireless network with unreliable links, the probability of a packet being successfully received by any node in a set of neighbors is greater than the probability of one specific node receiving it. In the context of routing across multiple wireless hops, there might therefore be a benefit to having more than one candidate next hop from each node to a given destination. The likelihood that a packet transmission is received by one of these candidates is higher than if we have a single next-hop.

The key question is then to decide, at each node, which neighbors are admissible as relays for a packet. This decision must be guided by the following tradeoff: on the one hand, by increasing the number of candidate relays, we decrease the cost to send to any of these relays. On the other hand however, some of these candidate relays may not be as well positioned as the next-hop along the shortest path to the destination, and so by becoming less and less “picky” about our candidate relays, we increase the likelihood of a packet veering away from the shortest-path route, and ultimately we may even introduce loops in our routing topology.

Biswas and Morris proposed the ExOR [1] routing protocol, that increases throughput of 802.11b multi-hop wireless networks by choosing the effective next-hop relay after the packet has been transmitted. In this work the choice of effective relay is driven by single-path routing metrics. In other words, the effective relay is chosen to be the receiver with shortest single-path distance to the destination. If no receiver has lower single-path distance, the sender retransmits.

While such a strategy has the advantage of leveraging well-understood single-path concepts, it does not result in the optimal routing choices. The reason is that shortest single-path route costs do not reflect route costs when using anycast forwarding. Consider for example the network of Figure 1. If the source can only use relays that are closer in single-path distance to the destination (as in ExOR), then it will only be able to use as next-hop its bottom-most neighbor, because any of the source’s three neighbors in the dense mesh are further in single-path distance to the destination as in ExOR, then it will only be able to use as next-hop its bottom-most neighbor, because any of the source’s three neighbors in the dense mesh are further in single-path distance to the destination as the source is.

In contrast, this paper shows that a different type of cost metric and routing structure should be used, which we call shortest anypath routing. ExOR focuses on throughput and the expected transmission-count (ETX) metric. We show in addition how anycast forwarding can be useful in low-power settings, for example to reduce the energy cost or delay of packet forwarding. In summary, the contributions of this paper include:

- Defining a framework for anypath routing that includes the notions of anycast link cost, anypath route, and anypath route cost,
- Formulating an algorithm that provably computes the shortest anypath route between each node and a destination, and can be implemented in a distributed setting,
- Defining a link cost criterion that must be respected for shortest anypath routes to be loop-free, and
- Introducing novel anycast forwarding mechanisms with which anypath routing can be used to improve energy-

---

Fig. 1. Mis-match of single-path metrics with anypath 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.
efficiency and latency of low-power wireless networks.

The paper proceeds as follows. We review some related work in the next section. In Section III, we introduce our framework for anycast forwarding and its associated concepts and notions of cost. Section IV defines anypath routes, their cost, and shows an algorithm to compute the shortest anypath routes. In Section V, we show new ways to exploit anycast forwarding with low-power wireless link layers. In Section VI we give numerical results, and finally we conclude in Section VII.

II. BACKGROUND AND RELATED WORK

The general idea of link-layer anycasting has been previously proposed and motivated in various forms. Larsson [2] proposed a joint forwarding and MAC 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 relay. Choudhury and Vaidya [3] 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. Our framework accommodates both of these approaches, that we call respectively receiver-driven and sender-driven anycast forwarding (Section V-B.1).

The aforementioned works focus on link-layer mechanisms to implement anycast forwarding. They assume that the network layer maintains a list of possible next-hop candidates (e.g., with a multi-path routing protocol), that is provided to the link layer, and do not propose specific strategies for the selection of these candidates by the routing protocol.

Jain and Das [4] go a step further by integrating an anycast extension of the 802.11 MAC layer with the multi-path AODV (AOMDV) [5] routing protocol. They observe the same tradeoff as [3] between number of candidates and path length, and modify AOMDV to allow the use of paths up to one hop longer than the shortest path. This choice is based on an empirical evaluation, the details of which are not given.

Note that the original design goal of most multipath routing protocols (and in particular, of AOMDV) is usually to improve load-balancing, redundancy or failover by providing multiple route choices. This is in contrast with anypath routing that provides multiple next-hop in order to take advantage of anycast forwarding.

One example of multipath routing in a wireless network is the work of Srinivas and Modiano [6], who propose an algorithm that finds minimum-energy $k$ link- or node-disjoint between two nodes. For link-disjoint paths, a node can have multiple outgoing edges, and their scheme takes into account the energy savings that are realized in this case thanks to the broadcast nature of the wireless medium. In the context of wired networks, another example is the work of Zaumen and Garcia-Luna-Aceves [7]. 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 anypath route in Section IV is similar to theirs, but our notion of shortest anypath routes is different, because our cost model is designed to reflect the use of anycast forwarding.

This work is not the first to consider anypath routing in the context of low-rate wireless sensor networks. Parker and Langendoen evaluated Guesswork, a protocol similar to ExOR in simulation using existing low-power link protocols [8] [9] [10]. They do not modify these link protocols however to specifically take advantage of anycast forwarding.

Finally, Zhong et. al. previously remarked [11] that the routes used by ExOR are not optimal. They introduce a metric that is similar to the remaining path cost remaining path cost defined in various forms in this paper, and is for the specific case of ETX [12] costs. They also propose a heuristic-based method that starts from the same routes as ExOR (e.g., based on single-path metrics), and then prunes certain candidate relays. The goal is to find smaller candidate relay sets than ExOR, in order to reduce arbitration overhead. In contrast, the shortest anypath routes computed here often have more candidate relays, as illustrated in the experimental part of this paper; furthermore arbitration overhead is not a critical issue with the anycast forwarding schemes developed here, as we discuss in Section V-C. In further contrast the algorithm proposed here is not based on single-path routing and computes provably shortest anypath routes.

III. ANYCAST FORWARDING

In this section, we introduce a framework encompassing anycast forwarding and its associated costs. Each node $i$ has a candidate relay set (CRS), denoted $C(i)$ (or $J$), containing all the nodes which may be used as relays for packets forwarded by $i$ toward the destination. For clarity, we assume in this paper that the destination is node 1; anycast routing applies of course with general traffic patterns and more than one destination.

The anycast link cost (ALC) $d_{i,j}$ is 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. Similarly to standard unicast link costs, choosing an anycast link cost is a modelling decision that depends on the cost criterion of our network. Anypath routing is worthwhile with anycast link costs that decrease when the candidate set is expanded; otherwise there is no advantage to having more than one candidate relay.

An ALC should have the following two simple properties:

The first is that it reduces to the unicast link cost for a CRS of size 1:

Property 1: If $J = \{j\}$, then $d_{i,j} = d_{ij}$.
In other words the anycast link cost is a generalization of unicast link costs. The second property is that the ALC to a set containing the sender node is 0; this property generalizes the property \( d_{i1} = 0 \) of unicast distances.

Property 2: If \( i \in J \), then \( d_{iJ} = 0 \).

A. Always-On Anycast Link Costs

We now give two simple examples of anycast link costs. Let us first define the anycast link delivery probability, which generalizes the unicast link probabilities \( p_{ij} \) to the anycast case. The probability that a unicast packet from \( i \) is received by \( j \) becomes the probability that an anycast packet from \( i \) is received by \( at least one \) node in the set of nodes \( J \). We denote this probability \( p_{iJ} \), and its expression is:

\[
p_{iJ} = (1 - \prod_{j \in J} (1 - p_{ij}))
\]

1) Transmission-count metric (ETX): The ETX metric [12] counts the expected number of transmissions to successfully deliver a packet across an unreliable link using link-layer retransmissions. With anycast forwarding, the ETX is generalized to the expected number of transmissions until at least one node in \( J \) receives the packet. Its expression is:

\[
d_{ij}^{ETX} = \frac{1}{p_{ij}}
\]

2) Delivery probability metric (E2E): Another possible anycast link cost simply considers the probability of successful packet delivery. This anycast link cost is defined as negative logarithm of \( p_{iJ} \), so that the costs can be added across several anycast links to obtain the end-to-end delivery probability:

\[
d_{ij}^{E2E} = -\log p_{ij}
\]

Note that \( p_{iJ} \) increases for every node that is added to \( J \), and thus so do the E2E and ETX anycast link costs.

B. Effective Relay Selection

If multiple nodes receive a packet, we must select one (or possibly many) of these nodes to be the next relay. This choice is made by the effective relay selection (ERS) policy. In this paper we exclusively consider the ERS-best policy, where the node with lowest distance to the destination among all receivers is selected as the next-hop. The notion of distance here is of course key; as discussed with Figure 1 different distance metrics will lead to different choices. The key is of course to have a distance metric that captures the true cost to reach the destination from each node.

Other policies are possible; for example with ERS-any a node is chosen at random among the receivers to become the relay, and with ERS-all all receivers relay the packet (thus creating duplicates). These policies are discussed in [13] and we do not cover them here; however we note that our framework can accommodate these and other policies simply by changing the expression of remaining path cost to appropriately reflect the ERS policy employed.

Note that the ERS strategy used to select the effective relay is a matter of policy and is conceptually separate from the coordination mechanism used to implement this decision in a distributed setting. We call this mechanism relay arbitration, and discuss it in Section V-C.

C. Remaining Path Cost

With unicast forwarding, it is trivial that the remaining cost for a packet to reach the destination after we have forwarded it is the path cost from next-hop to the destination. With anycast forwarding, there is more than one possible next hop, 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 \( J \) of node \( i \). Denote by \( D_k \) the cost to reach the destination from a node \( k \). (This distance may be for example the single-path distance to the destination, or the anypath distance that introduced in the following section).

If \( D_k = D \) for all \( k \in J \), then clearly \( R_{iJ} = D \). It is equally clear that if all nodes in \( J \) receive all packets from \( i \), then \( R_{iJ} = \min_{k \in J} D_k \). However, in the general case each node \( k \) in \( J \) receives the packet with some probability \( p_{ik} \). Assume wlog that the nodes in \( J \) are sorted by their distance to the destination, i.e., that \( D_1 < D_2 < \ldots < D_j \). Then the remaining path cost is:

\[
R_{iJ}^{\text{best}} = \frac{1}{1 - \prod_{k \in J} p_{ik}} \left( p_{i1} D_1 + \sum_{j=2}^{n} p_{ij} D_j \left( \prod_{k=1}^{j-1} p_{ik} \right) \right),
\]

with the following simplification when \( p_{ik} = p \):

\[
R_{iJ}^{\text{best}} = \frac{p}{1 - (1-p)^n} \sum_{j=1}^{n} (1-p)^{j-1} D_j,
\]

where we have used the superscript \( \text{best} \) to emphasize that the above expressions of remaining path cost are valid in conjunction with the policy ERS-best.

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 candidate relay set \( J \) 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 relay.

IV. SHORTEST ANYPATH ROUTING

We first clarify exactly what is meant by an anypath route. We define an anypath route \( R \) from a source to a destination to be a 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. An acyclic anypath route is an anypath route that contains no cycles. We illustrate this definition with the example of Figure 2, where the black nodes and links form an anypath route between the source and destination. By connecting node \( A \) as shown we cease to
have an anypath route since \(A\) is not a successor of the source. Similarly, node \(B\) is not a predecessor of the destination, so by adding it we also cease to have an anypath route.

### A. Cost of Anypath Routes

Having defined anypath routes, we now need a definition of their cost.

1) **Cost of a walk in an anypath route:** At each use of an anypath route, a packet may traverse a different sequence of nodes, since there are (in general) multiple possible relays at each hop.

A walk \(W\) in an anypath route \(R\) 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 \(R\). We use the term “walk” and not “path” because a node can be visited more than once if the anypath route is cyclic. We can now define the cost of a walk relative to the anypath it traverses.

**Definition 1 (Cost of a walk in an anypath route):** Let \(W = (s, n_1, n_2, \ldots, n_k, 1)\) be a walk in anypath route \(R\). The cost of \(W\) relative to \(R\), denoted \(c(W|R)\), is the sum of the anycast link costs in \(R\) of the nodes in the path \(W\):

\[
c(W|R) = \sum_{i \in W} d_{iC(i)} = d_{sC(s)} + d_{n_1C(n_1)} + d_{n_2C(n_2)} + \ldots + d_{n_kC(n_k)}.
\]

It is important to emphasize that the cost of a walk depends on the anypath route \(R\) that it traverses, as illustrated in the following example.

**Example 1 (Cost of a walk depends on traversed anypath):** We illustrate this dependence in Figure 3, where we compute the cost of the same path \(W = (a, b, c, d)\) relative to four traversed anypath routes. All links have delivery probability 0.5, and the cost metric is ETX. In Figure 3(a), node \(a\) has two candidate relays, and so its ALC is \(d_{aC(a)}(1-0.5^2)^{-1} = 4/3\). Nodes \(b\) and \(c\) have a single candidate relay and have ALC equal to 2, giving a walk cost \(c(W|R) = 5.33\). In Figure 3(c), the anypath route is different than in (a), but the cost of the walk is the same, because anycast link costs are not affected by the additional incoming links. In Figure 3(b) the costs at both nodes \(b\) and \(c\) are lower due to their having additional candidate relays. Finally in Figure 3(d), the anypath route is equal to the walk itself, and the cost of the walk is the same as the cost of the single-path route from \(a\) to \(d\).

2) **Anypath route cost:** Of course there are multiple possible walks for a packet to go from the source to the destination in an anypath route. Each possible walk \(W\) happens with some probability \(P(W)\). It is then natural to define the cost of an anypath route cost as the expected cost of traversing it:

**Definition 2 (Anypath route cost):** The cost of an anypath route \(R\) is the expected cost of a walk across that route.

\[
Cost(R) = \sum_{W \in R} P(W) \cdot c(W|R),
\]

where the sum is over all possible walks from the source to the destination of \(R\).

A few remarks on the anypath route cost:

- The probability of a path \(W\) depends on the choice that is made by the ERS policy at each anycast forwarding phase. Thus, it depends on costs of the nodes in the route, since when multiple nodes receive a packet, the ERS policy chooses the effective relay based on their cost to the destination. Since the cost of an anypath route depends on the costs from interior nodes in that route, we must therefore know the costs of the interior nodes in order to compute the cost of the route.
- The anypath route cost generalizes the cost of a single path route: if the anypath route is a single path, its cost is the sum of its constituent unicast link costs.
- The anypath route cost is based on anycast link costs, that are used to compute the cost of each walk. The remaining path cost is absent from the definition.

### B. Physical Cost Models

The benefit of anycast forwarding is that in many cost models, it is less costly to transmit to any node in of a set of neighbors than to a single neighbor. By how much can the forwarding cost decrease with anycast forwarding? We answer this question with the definition of a physical cost model.

Consider a node \(i\) with a given candidate relay set \(J\) and a cost \(D_i\) to the destination. Consider now a neighbor \(k\) of node \(i\) with greater distance to the destination \(D_k > D_i\). Can node \(i\) decrease its cost to reach the destination by adding \(k\) to its candidate relay set? If the answer is positive, then we have following (somewhat counterintuitive) situation: we can...
reduce the cost of sending a packet from i to the destination by going through a relay that has a higher cost to reach the destination than i itself! This apparent paradox motivates the definition of a physical cost metric, with which such situations are impossible.

Definition 3 (Physical cost model): Consider a node i with candidate relay set 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 \( J' = J \cup k \). A physical cost model is one for which

\[
d_{ij} + R_{ij} \geq d_{ij} + R_{ij},
\]

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

We now return to the two ALCs defined in Section III.

Property 3: The expected transmission count (ETX) anycast link cost is a physical cost model.

Property 4: The end-to-end delivery probability (E2E) anycast link cost is not a physical cost model.

These two properties are proven in the Appendix. To understand why E2E is not a physical metric, consider a sender i transmitting a packet. Assume that only one neighbor, node j, receives this packet, and that the packet delivery probability from j to the destination is lower than from i. The delivery probability for packets originated by i increases if we allow j as a relay, even though j itself has lower delivery probability than i.

C. Shortest Anypath Routes

Now that we have defined the cost of an anypath routes, the shortest anypath route \( R \) between two nodes is defined naturally as the anypath route with lowest possible cost. Similarly, the shortest acyclic anypath route is the acyclic anypath route with lowest cost.

Property 5: The cost of the shortest anypath route between two nodes is either smaller than or equal to shortest single-path cost between two nodes.

The proof of this property, given in the Appendix, is fairly obvious once we consider that the set of all anypath routes between two nodes includes all single-path routes.

1) Presence (or absence) of cycles in shortest anypath routes: Our definition of anypath routes allows the presence of cycles. Is this necessary, and in particular, can a shortest anypath route ever contain a cycle? With a physical cost model, the shortest anypath route is always acyclic, but with non-physical cost model the shortest anypath route may contain cycles.

Proposition 1: In a physical cost model, no cyclic anypath route has lower cost than the shortest acyclic anypath route.

The proof of this proposition (given in the Appendix) proceeds by showing that with a physical cost, it is always possible to obtain a lower-cost acyclic anypath route from a cyclic anypath route.

Corollary 1: An algorithm to find shortest anypath routes under a physical cost model need not consider cyclic routes.

This is fortunate, since reasoning about the subset of anypath routes that are acyclic is easier than reasoning about all possible anypath routes. Note however that in a non-physical cost model, the shortest anypath route may contain cycles! In the case of the non-physical E2E metric, one way to interpret why the shortest anypath route can contain cycles is to see that this metric captures the probability of end-to-end delivery for links without retransmissions. While it is preferable for a packet to make progress to the destination at every hop, it is preferable (from the perspective of delivery probability) to allow a hop that moves away from the destination (and thus may lead to a loop) than to lose the packet altogether. In this sense, any path routing with a non-physical cost metric is quite reminiscent of deflection routing or hot potato routing.

D. Computing Shortest Anypath Routes with Physical Costs

We now show an algorithm to compute shortest anypath routes. We assume throughout this section the use of a physical cost model; therefore we have from Proposition 1 that shortest anypath routes in this section are acyclic.

The algorithm is based on the classical Bellman-Ford algorithm, and its development bears some resemblance with the development for the single-path case [14]. The upper bound on convergence time on the algorithm’s convergence (in number of iterations) is the same for any path routing as for single-path routing. Not surprisingly the complexity of the anypath algorithm is however greater, because it must compute at each iteration a set \( C(i) \subseteq N(i) \) for each node and not just a single next-hop.

How does a node select which of its neighbors should be candidate relay nodes? With anypath routing, the expression to minimize becomes the sum of the anypath link cost and the remaining path cost \( D_i = d_{ij}C(i) + R_{ij}C(i) \), and this sum must be minimized over all possible subsets \( J \subseteq N(i) \):

\[
D_i = \min_{J \subseteq N(i)} [d_{ij} + R_{ij}],
\]

This is called the Bellman’s Equation for anypath routing, and it represents the steady-state of the anypath Bellman-Ford algorithm that we now describe.

We compute the shortest anypath distances (and corresponding routes) iteratively as follows. At each iteration, we update the value \( D^h_i \) at each node i, where \( h \) is the iteration index. This \( D^h_i \) is the shortest-anypath distance estimate from i to the destination at the \( h \)-th iteration, and we shall show that the sequence converges toward the shortest-anypath distance \( D_i \). By convention, we take:

\[
D^0_i = 0, \quad \text{for all } h,
\]

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

\[
D^{h+1}_i = \min_{J \subseteq N(i)} [d_{ij} + R^{h}_{ij}] \quad \text{for all } i \neq 1,
\]

where \( R^{h}_{ij} \) is the remaining path cost computed using the distances \( D^h_j, j \in J \) from the previous iteration. The candidate relay set 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:

\[ D_i^0 = \infty, \quad \text{for all } i \neq 1. \]

The algorithm terminates when:

\[ D_i^h = D_i^{h-1}, \quad \text{for all } i. \]

In the following, a \((\leq h)\) anypath route is a route whose longest walk contains at most \(h\) hops. A shortest \((\leq h)\) anypath route from a node \(i\) is a shortest anypath route from \(i\) to the destination, subject to the constraint that the longest walk in the anypath route traverses at most \(h\) hops.

Proposition 2: The anypath Bellman-Ford algorithm computes, at iteration \(h\), the shortest \((\leq h)\) anypath distances from each node to the destination. Furthermore, the algorithm terminates after at most \(h^* \leq |N|\) iterations, and at termination, \(D_i^{h^*}\) is the shortest anypath distance from \(i\) to the destination.

The proof of this proposition, given in [15], proceeds similarly as for the single-path Bellman-Ford [14] algorithm. The main difference is that it requires first proving that every sub-anypath of a shortest anypath route is itself a shortest anypath route (this property is trivial in the case of single-path routes).

Note that just like single-path Bellman-Ford, the algorithm above can be implemented in a distributed setting, with nodes asynchronously recomputing their shortest-anypath distance (using eq. (8)) and advertising it to their neighbors.

V. APPLICATION TO LOW-POWER WIRELESS NETWORKS

Reduced energy consumption is a key objective in the design of protocols for many wireless networks, in particular wireless sensor networks. In many applications of wireless sensor networks, nodes must survive for extended periods (e.g., years) using a limited power source. With the radio being the dominant energy consumer [9] [16] of these devices, it is necessary to power it down whenever possible. This is called duty cycling, and we define the duty cycle \(\rho\) of a link-layer as the fraction of time it on average keeps the radio awake and listening. Of course a prerequisite to operating at low duty cycles is to have a low traffic rate; this is the case of many envisioned sensor network applications.

A. Two Models of Duty Cycling

Many strategies for low-power operation of wireless links have been proposed. In high-level terms, duty-cycling schemes trade off latency for energy efficiency, and a key practical difficulty to achieve low duty cycles (e.g., \(\rho < 10^{-2}\)) is to reliably rendezvous between a sender and a receiver, whose radio is turned off most of the time. Giving an exhaustive survey is beyond the scope of this paper. Instead, we define two models of duty cycling, based on two popular protocols used in wireless sensor networks. These models are intentionally simple and seek to capture the essential features of their respective protocols. We then use these models to show novel anycast forwarding mechanisms that improve energy efficiency and latency over the corresponding unicast duty-cycled link-layers.

In the models considered here, all nodes have the same duty cycle \(\rho\) and awaken to listen to the channel once within an interval of duration \(t_{rx}\). Note that in practice the duration of the listen state is lower-bounded by a minimal wakeup time, and hence decreasing \(\rho\) requires to increase \(t_{rx}\).

1) Synchronous duty cycling: In synchronous duty cycling, nodes know the sleep schedule of their neighbors. Schedules can be announced with beacons or piggybacked on packets. A sender waits until its neighbor is awake before transmitting a packet to that neighbor. Synchronous duty cycling can be clustered [10], in which case all nodes in a neighborhood coordinate to awaken at the same time, or individual [17], in which case each node maintains and advertises its own schedule. In either case, the average latency to send a packet to a neighbor increases with duty cycle \(\rho\).

2) Low-Power Listening: With low-power listening (LPL [9]), nodes do not keep track of each others’ duty cycles. As a consequence, they cannot simply start transmitting at the time when the destination wakes up, and so a packet is preceded by a long preamble that must last as at least long as the interval \(t_{rx}\) between node wakeups. The key advantage of these schemes over synchronous duty cycling is their robustness and simplicity; their drawback is the cost of transmitting long preambles.

B. Anycast Forwarding with Duty-Cycling

We now describe two new anycast forwarding mechanisms that improve the performance of the above duty-cycling schemes. Note that the use of the schemes presented here does not preclude the use of anycast forwarding to reduce the ETX (as in Section III); and both can be done in combination. However, to clearly highlight the specific features of each of the schemes below 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.

1) Duty cycling with individual synchronization: If a sender has multiple candidate relays, each with its own wakeup schedule, and if the sender knows these schedules, it can select the node that will wake up earliest as its next hop relay.

Assume that wakeup periods are uniformly distributed within the interval \(t_{rx}\). The delay for a node \(i\) to transmit to any node in \(J\) is therefore the minimum of \(|J|\) random variables that are uniformly distributed over \([0, t_{rx}]\). This minimum can be computed using order statistics [18], and is \(\frac{1}{|J|+1}\). Using this we can define the latency anycast link cost giving the expected delay to forward a packet to any node in a CRS:

\[ d_{LAT} = t_{pkt} + \frac{t_{rx}}{|J|+1}, \tag{9} \]

where \(t_{pkt}\) is the time for the transmission of the packet itself. In contrast, the average delay with unicast forwarding to hop is \(t_{pkt} + t_{rx}/2\).

Under the assumption of uniformly distributed wakeup schedules, the all nodes in a CRS \(J\) have equal probability
of being relays, and so the remaining path cost is:

\[ R_{i,j} = \frac{1}{|J|} \sum_{j=1}^{|J|} D_j. \]

Note that in this form of anycast forwarding, the next-hop decision is made by the sender, before transmission of a packet. We call this sender-driven anycast forwarding. In contrast, the next-hop decision can only be made after transmission for the other anycast forwarding mechanisms in this paper; we call these with receiver-driven forwarding.

2) LPL: We now show how anycast forwarding allows a sender to significantly reduce the total length of preambles transmitted when using LPL. The net effect is a reduction in energy cost and latency to transmit a packet.

The intuition is the following: Assume, as previously, that wakeup periods 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 normal unicast LPL, we have \( \lambda \geq 1 \). We say that a transmission hits a node if the preamble covers the node’s wakeup interval. Under our modelling decision (see Section V-B) of reliable links, the probability \( p_{hit} \) of hitting one specific neighbor will be \( \lambda \), but the probability of hitting any node in a CRS of size \( n \) will be

\[ p_{hit} = 1 - (1 - \lambda)^n. \] (10)

While \( p_{hit} \) increases with the size of the CRS, guaranteeing that some node in the CRS 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 \( 1/p_{hit} \).

Our cost metric here should reflect an entire transmission cost, which includes preamble transmission time as well as packet transmission time\(^2\). Note that there is a tradeoff between decreasing \( \lambda \) (the cost of a single transmission) and increasing \( 1/p_{hit} \) (the expected number of transmissions). The optimal point in this tradeoff depends on the size \( |J| \) of the CRS and the relative durations of \( t_{pkt} \) and \( t_{rx} \). We can now define the energy anycast link cost for LPL links with retransmissions:

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

where the numerator is the energy cost of one transmission, and is multiplied by the expected number of transmissions. Computing \( d_{i,j}^{LPL} \) 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 5. The optimal tradeoff point is for small values of \( \lambda \), (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 \( \lambda \), the transmission cost is reduced by a factor of 2.5 for “reasonable” CRS sizes (\( |J| \leq 10 \)). We point out that the gain over unicast LPL increases further with unreliable links, where retransmissions are needed not only due to preamble misses but also to link unreliability.

![Fig. 5. Anycast link cost](image)

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

C. Relay Arbitration

A large effort was devoted in previous work [1] [11] to designing efficient relay arbitration protocols, that coordinate nodes after a transmission so as to select the best relay when there are multiple receivers. Our focus in this paper is on anypath routes themselves and their computation. Still, we should point out that our the anycast forwarding for duty-cycling link layers presented here can get away with much lower relay arbitration overhead. In the case of anycast forwarding with synchronous duty cycling, this is because
the sender selects the next hop before transmission. In the case of anycast forwarding with LPL, Figure 5 shows that the optimal value of the preamble length $\lambda$ 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 exchange between multiple nodes.

VI. NUMERICAL RESULTS

We first evaluate the cost of paths found by anypath routing. We focus here on anypath routing with LPL and do not cover synchronous duty cycling due to lack of space. In this section we use the following terminology: $SP$ routes are shortest single-path routes as found by classical Bellman-Ford or Dijkstra algorithms. $AP$ routes are the shortest anypath routes found by the algorithm of Section IV. Finally, $SP-AP$ routes are anypath routes obtained using a single-path metric (such as in ExOR): nodes run a standard single-path algorithm and take as candidate relays all neighbors with lower single-path cost to reach the destination.

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 10 and networks with 500 nodes.

In our first set of simulations, we evaluate the cost of shortest anypath (SP) routes. For each node pair, we compute the shortest single-path route (using a standard Bellman-Ford algorithm) and we compute the shortest anypath route (using (8)). We then order nodes by single-path distance, and in Figure 6(a) plot the average shortest anypath distances as a function of this shortest single-path distance. The AP route costs are reduced by a factor between 1.8 and 2 compared to SP costs. Furthermore, the gap widens for diminishing duty-cycle $\rho$, 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 $\lambda$ as $t_{rx}$ is increased. Note that shortest anypath route costs are (roughly) a constant factor of single-path costs; the gain of anypath routing increases with density rather than diameter.

We also compared AP route costs with those of SP-AP routes. These results are superimposed in Figure 6(a). The cost of the SP-AP routes is approximately 40% higher than that of shortest anypath routes (AP). Note that our networks have uniformly distributed nodes and are quite homogeneous; the gap between shortest anypath routing and SP-AP anypath routing will widen with less homogeneous topologies, where situations like the one of Figure 1 become more frequent.

One way to characterize an anypath route is to consider the number of candidate relays that nodes in this route have. On the one hand, anycast forwarding costs decrease with large CRS sizes. On the other, using increasingly large CRS sizes means that a packet may wander away from the most direct path. Do shortest anypath routes have more candidate relays than ExOR-like routes, where candidate relays are simply those nodes with lowest single-path distance? Or do shortest anypath routes have similar CRS sizes, but have improved performance due to the choice of relay being informed by the more suitable ALC metric? We define the average out-degree as the empirical average of $|C(i)|$ for nodes at a given single-path distance to the destination, and plot it in Figure 6(b), averaged over 10000 network realizations. This shows us that, at least under the $d_{ij}^{AP}$ cost model, AP routes are able to use more candidate relays than SP-AP routes; nodes in AP routes have about 4 candidate relays, in comparison with 2 for SP-AP routes. An example using a simulated network of 20 nodes is given in Figure 7.

Route costs are a primary measure of a routing algorithm’s performance, but are not the only measure. Robustness is another important aspect of any algorithm that is intended to run in a distributed wireless setting. One very elementary notion of robustness is the resilience of routes in the face of topology changes. We studied this form of robustness by running the following experiments. First we generate a network realization, and compute all shortest path routes in it. Then, we randomly remove a number of links in this network. Links are independently removed with probability $p_c$, 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 routes; for AP or SP-AP routes disconnectedness means that there is no path to reach the destination.

We plot the empirical CDF of $N_d$ in Fig. 6(c). Of course, any routing algorithm with multiple paths is more robust than single-path routing, and so it is expected that AP routing has fewer disconnections than SP. More interesting however is that AP routes are also significantly less prone to disconnection than SP-AP routes. For example, the probability that less than 10% of routes are disconnected with AP is over 0.95, while it is only 0.65 with SP-AP. This is a direct consequence of
AP’s larger candidate relay sets as shown in Figure 6(b). Note that robustness is a multi-faceted and general property, and we do not claim that the results shown here constitute a complete evaluation of anypath routing’s robustness. Beyond the binary notion of disconnectedness, a full investigation should also examine the cost of routes that are computed with an approximate (or noisy) view of network topology, as is the case in wireless networks. Our intuition is that integrative nature of the anypath cost metric should make anypath routes more stable to such “noisy” inputs than SP or SP-AP.

VII. Conclusion

We have introduced a new algorithm to compute shortest anypath routes in multi-hop wireless networks, and introduced new ways to exploit anycast forwarding in the context of low-power wireless networks. We are in the process of evaluating this algorithm on a wireless sensor network testbed using TinyNodes [16]; initial results confirm the practicality and performance of shortest anypath routing.

APPENDIX

A. Proof of Property 3 (ETX is physical)

To lighten notation, we write \( p_k \) instead of \( p_{ik} \) in the following, given that we are only considering links from \( i \) to a neighbor. Consider a node \( i \) with candidate relay set \( J = \{j\} \). Under the ETX metric and under the ERS-best model, the cost \( D_i \) is determined by (2) and (4), and is \( D_i = \frac{1}{p_{ij}} + D_j \). Now add a node \( k \) with \( D_k \geq D_i \) to the candidate relay set, to obtain a set \( J' = \{j, k\} \), and examine the distance \( d_{i,j'} + R_{i,j'} \). From (2) we have the anycast link cost from \( i \) to \( J' \):

\[
d_{i,j'} = \frac{1}{1 - (1 - p_j)(1 - p_k)},
\]

and from (4) we have the remaining path cost:

\[
R_{i,j'} = \frac{1}{1 - (1 - p_j)(1 - p_k)}(p_j D_j + (1 - p_j)p_k D_k),
\]

where we have used the hypothesis that \( D_k < D_j \).

We now consider the sum of these two quantities and show that it is greater than \( D_i \):

\[
d_{i,j'} + R_{i,j'} = \frac{1}{1 - (1 - p_j)(1 - p_k)}(1 + p_j D_j + (1 - p_j)p_k D_k)
\]

\[
= f(p_k), \quad p_k \in [0, 1]
\]

\[
\geq f(0)
\]

\[
= \frac{1}{p_j} + D_j
\]

\[
\equiv d_{i,j} + R_{i,j} = D_i,
\]

where we used in (a) the fact that \( 0 < p_k < 1 \), and the quantities \( p_j, (1 - p_j) \), and \( D_j \) are positive, and in (b) the definition of \( D_i \).

We have not yet shown that the expected transmission count ALC is a physical cost model, since the development above considers only the special case where \( |J| = 1 \). Let us now consider the general case where the candidate relay set has an arbitrary number of nodes: \( |J| = n \). Consider now another candidate relay set \( H \) consisting of a single node \( h \), such that \( p_h = p_{i,j} = (\prod_{j \in J}(1 - p_{ij})) \), and \( D_h = R_{i,j} \) (where \( R_{i,j} \) is computed according to (4)). Observe that \( d_{i,j} = d_{i,H} \) and \( R_{i,j} = R_{i,H} \), and that it is equivalent to consider either the candidate relay set \( J \) or our constructed set \( H \). Therefore, any general case \( J \) can be reduced to a singleton CRS, a situation for which the above development has shown that the physical cost property holds.

Finally, we have only considered the ERS-best policy in the above development. Note that with any other policy, the remaining path cost when adding node \( k \) is greater than with ERS-best, and so the fact that the expected transmission count ALC is a physical cost metric for ERS-best implies that it is a physical cost metric for any ERS.

B. Proof of Property 4 (E2E not physical)

We prove that the E2E metric is not physical by using a counter-example. Consider the network of Figure 8, where node 1 is the destination. The link from \( i \) to the destination has delivery probability \( p_{i,1} = \frac{1}{2} \), the link from \( k \) to the destination has delivery probability \( p_{k,1} = \frac{3}{4} \), and the link between \( i \) and \( k \) has probability \( p_{ik} = \frac{1}{2} \). In Figure 8(b), node \( i \) has as candidate
we should have

However, using equation (3) we can compute that

$j$

This candidate relay set the singleton set containing the destination only, i.e., $J = \{0\}$. The remaining path cost is therefore $R_{i,j} = 0$, and we have $D_1 = d_{i,j} + R_{i,j} = -\log 2$. Node $k$ has the same candidate relay set consisting of the destination only, giving us $D_k = -\log 2 > D_1$.

Consider now Figure 8(c), where node $i$ uses as CRS the set $J' = \{0, k\}$. If the delivery probability ALC were a physical cost model, the cost at node $i$ would increase with this candidate relay set $J'$, because $D_k > D_1$. More precisely, we should have $d_{i,j'} + R_{i,j'} \geq d_{i,j} + R_{i,j}$, where $J = \{0\}$. However, using equation (3) we can compute that

$$d_{i,j'} = -\log (1 - (1 - p_{i1})(1 - p_{ik})) = -\log \frac{3}{4},$$

and

$$R_{i,j'} = -\log ((1 - (1 - p_{i1})(1 - p_{ik})) (p_{i1} + (1 - p_{i1})p_{ik}p_{k1}) > 0.$$

Putting both together, we obtain

$$d_{i,j'} + R_{i,j'} = -\log \frac{3}{4} - \log \frac{3}{4} < -\log \frac{1}{2} = d_{i,j} + R_{i,j},$$

showing that the delivery probability ALC is not a physical cost model.

Since this example is simple, we can also directly compute the raw probabilities. A packet sent by $i$ arrives successfully at the destination if either it is received directly by the destination, or it is successfully relayed by node $k$ to the destination. The probability $p$ of successful delivery is thus:

$$p = p_{i1} + (1 - p_{i1}) p_{ik} p_{k1} = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4} = \frac{9}{16}.$$ 

Of course this “direct” computation gives exactly the same result as the development above; we can verify that $e^{-(d_{i,j'} + R_{i,j'})} = \frac{9}{16}$.

C. Proof of Proposition 1

By contradiction. Assume that the shortest anypath route $R$ contains at least one node that is not acyclic. It therefore contains at least one cycle. Let us consider one cycle $(n_1, n_2, n_3, \ldots, n_{k-1}, n_1)$, with nodes having destination distances $(D_1, D_2, D_3, \ldots, D_{k-1}, D_1)$. Since this is a cycle, one of two cases must be true: either $D_1 = D_2 = D_3 = \ldots = D_{k-1}$, or there are two consecutive nodes $(i, j)$ in the cycle with $D_i < D_j$.

We consider each of these two cases individually.

- **Case 1**: $D_1 = D_2 = D_3 = \ldots = D_{k-1}$. No cycle can contain the destination, since the destination has no successor. Therefore any cycle in an anypath route cannot be closed; in other words there must be at least one node in $(n_1, n_2, n_3, \ldots, n_{k-1}, n_1)$ with a successor that is not in the cycle. Let us call this node $n_i$; $n_i$ has one successor $n_{i+1}$ in the cycle, and a set of other successors $K$ (with $|K| \geq 1$). (In the terminology used previously this amounts to saying that $C(n_i) = \{n_{i+1}\} \cup K$). Now, if we consider the definition (Definition 3) of a physical cost model, and set $J = K$ and $J' = \{n_{i+1}\} \cup K$, we see that we can remove $n_{i+1}$ from node $n_i$’s successors (i.e., set $C(i) = K$) and reduce the cost $D_i$.

- **Case 2**: $D_i < D_j$ for two consecutive nodes $(i, j)$ in the cycle. Since anycast link costs are positive, node $i$ must have another successor(s) in addition to $j$ (or else we would have $d_{i,j} = d_{i,j'} + R_{i,j'} = d_{i,j} + R_{i,j'} > D_j$). Again, considering the definition (Definition 3) of a physical cost model shows that the cost $D_i$ can be reduced by removing $j$ from node $i$’s candidate relay set.

Returning to our shortest anypath route $R$, let us construct the anypath $R'$ by applying either of the two modifications above to every cycle in $R$. Call $D'_k$ the distance from node $k$ to the destination, we have that $D'_k \leq D_k$ for all nodes, with the inequality being sharp for at least one node (the one whose candidate relay set was pruned according to the modifications above). Since this example is simple, we can also directly compute the raw probabilities.

In consequence, all possible walks have either lower or equal cost over $R'$ than over $R$, with at least one having a lower cost (i.e., a walk going through the node whose candidate relay set was pruned). The constructed route $R'$ has lower cost than $R$, contradicting our initial assumption.

D. Proof of Proposition 2

We must first prove an auxiliary but important lemma.

**Proposition 3**: Let $R$ be a shortest anypath route from a source to a destination, and node $k$ be an interior node in $R$. Call $R_k$ the sub-anypath route of $R$ from node $k$, and define $D_k = \text{Cost}(R_k)$. Then,

$$D_k = D'_k,$$

where $D'_k$ is the shortest anypath distance from node $k$ to the destination.

**Proof**: Call $T$ the shortest anypath route from node $k$ to the destination. We have therefore $\text{Cost}(T) = D'_k$. Since $T$ is
the shortest anypath from $k$ to the destination, we cannot have $D_k^* > D_k$, or otherwise $R_k$ would be a shorter anypath than $T$. 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 shortest anypath route $R$ that $R_k$ is a subanypath of. If $D_k^* < D_k$, then any packets arriving at $k$ from the source of route $R$ toward the destination can be forwarded using $T$. This results in a new route that we call $R^*$, going between the same source and the destination as route $R$. To complete the proof we observe that $R^*$ has lower cost than $R$, contradicting our initial assumption that $R$ was a shortest anypath route.

We now return to the proof of Proposition 2. We prove the first part of the proposition by induction over $h$.

**Case $h = 1$**. Using (8) and our initial conditions, we have

$$D_i^1 = d_{i1},$$

for all $i 
eq 1$, which is indeed the shortest $(\leq 1)$ anypath distance to the destination.

**Induction over $h$**. We assume that $D_i^h$ is equal to the shortest $(\leq h)$ anypath distance from $i$ to 1, and must show that $D_i^{h+1}$ is equal to the shortest $(\leq h + 1)$ anypath distance. There are two possible cases for each node $i$. The first is that the shortest $(\leq h + 1)$ anypath route from $i$ to 1 contains a longest walk with $h$ or less hops. We call this route $R_i^h$, and in this case we have $Cost(R_i^h) = D_i^h$. The second possible case is that the shortest $(\leq h + 1)$ anypath route from $i$ to 1 contains a longest walk with $h + 1$ hops. Call this route $R_i^{h+1}$. It has cost

$$Cost(R_i^{h+1}) = d_{iC(i)} + R_{iC(i)}$$

This route consists of $|C(i)|$ links from $i$ to each node in its CRS $C(i)$, and then of $|C(i)|$ sub-anypath routes from each node in $C(i)$ to 1 that each have a $h$-hop longest walk. From Proposition 3, we know that these sub-anypath routes must be shortest anypath routes. Given this structure, there is no possible candidate relay set among $i$’s neighbors that has a lower cost to reach the destination with $(\leq h)$ walks:

$$Cost(R_i^{h+1}) = d_{iC(i)} + R_{iC(i)} = \min_{j \in 2^{N(i)}}[d_{ij} + R_{ij}^h] = D_i^{h+1}.$$\text{(4)}

Calling $S_i^{h+1}$ the shortest $(\leq h + 1)$ anypath route length from $i$ to 1, these two cases thus give:

$$S_i^{h+1} = \min \{Cost(R_i^h), Cost(R_i^{h+1})\} = \min \{D_i^h, \min_{j \in 2^{N(i)}}[d_{ij} + R_{ij}^h]\} = \min \{D_i^h, D_i^{h+1}\} = D_i^{h+1},$$

and so $D_i^{h+1}$ is the shortest anypath distance from $i$ to 1.

The second part of the proposition follows simply from the first part and the fact that in a network with $|N|$ nodes, the longest possible path has at most $|N| - 1$ hops.