

Valuable Detours: Least-Cost Anypath Routing

Henri Dubois-Ferrière, Matthias Grossglauser, and Martin Vetterli, *Fellow, IEEE, ACM*

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. This observation was previously exploited by opportunistic routing protocols by using single-path routing metrics to assign to each node a group of candidate relays for a particular destination. This paper addresses the *least-cost anypath routing (LCAR)* problem: how to assign a set of candidate relays at each node for a given destination such that the expected cost of forwarding a packet to the destination is minimized. The key is the following tradeoff: On one hand, increasing the number of candidate relays decreases the forwarding cost, but on the other, it increases the likelihood of “veering” away from the shortest-path route. Prior proposals based on single-path routing metrics or geographic coordinates do not explicitly consider this tradeoff and, as a result, do not always make optimal choices. The LCAR algorithm and its framework are general and can be applied to a variety of networks and cost models. We show how LCAR can incorporate different aspects of underlying coordination protocols, for example a link-layer protocol that randomly selects which receiving node will forward a packet, or the possibility that multiple nodes mistakenly forward a packet. In either case, the LCAR algorithm finds the optimal choice of candidate relays that takes into account these properties of the link layer. Finally, we apply LCAR to low-power, low-rate wireless communication and introduce a new wireless link-layer technique to decrease energy transmission costs in conjunction with anypath routing. Simulations show significant reductions in transmission cost to opportunistic routing using single-path metrics. Furthermore, LCAR routes are more robust and stable than those based on single-path distances due to the integrative nature of the LCAR’s route cost metric.

Index Terms—Cross-layer design, routing protocols, wireless mesh networks.

NOTATION AND ACRONYMS

$N(i)$	Neighbors of node i .
p_{ij}	Packet reception probability from i to j .

Manuscript received March 12, 2009; revised February 13, 2010; accepted July 15, 2010; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Shakkottai. Date of publication September 07, 2010; date of current version April 15, 2011. This work was supported in part by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under Grant 5005-67322. The results presented in this paper are derived from a 2006 Ph.D. dissertation by H. Dubois-Ferrière at the EPFL, Lausanne, Switzerland; a subset of these results was presented in an earlier paper at the Allerton Conference on Communications, Control, and Computing, Monticello, IL, September 26–28, 2007.

H. Dubois-Ferrière was with the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015, Switzerland. He is now with Riverbed Technology (e-mail: henri.dubois-ferriere@a3.epfl.ch).

M. Grossglauser is with the Nokia Research Center (NRC), Helsinki, Finland, and also with the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015, Switzerland (e-mail: matthias.grossglauser@epfl.ch).

M. Vetterli is with the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015, Switzerland, and also with the University of California, Berkeley, Berkeley, CA 94720 USA (e-mail: martin.vetterli@epfl.ch).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNET.2010.2070844

$J(i)$ or J	Candidate relay set (CRS) at node i .
$d_{i,J}$	Anycast link cost (ALC) from i to J .
$R_{i,J}$	Remaining path cost (RPC) from J to the destination.

I. INTRODUCTION

IN WIRELESS networks, it is often less costly to transmit a packet to *any* node in a set of neighbors than to one specific neighbor. 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)* [3]–[7].

The starting point of this work is the following question: With opportunistic routing, are there practical and general ways to: 1) select the optimal¹ set of candidate relays that can be used at each node to reach a given destination: and 2) prioritize these relays in order to select the optimal effective forwarder when multiple candidate relays have received a packet?

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

In addition, the prioritization of relays must take into account a second tradeoff, linked to the underlying link-layer coordination protocol: As the number of candidate relays grows, the forwarding cost decreases, but the overhead of link-layer coordination increases because the number of nodes participating in the coordination protocol is higher. Having more candidate relays also increases the risk of duplicate transmissions, where more than one receiving relay mistakenly forwards a packet.

With opportunistic routing, each packet can traverse a multitude of possible paths to reach a destination, with each path having a different cost. Which path each packet follows depends on the nondeterministic 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 relays gives rise to a probability distribution over all possible paths between the source and destination, and this distribution determines the expected cost of using a route.

¹Of course, the notion of optimality is relative to the model of the network, and any routing algorithm can only be as good as the model and input metrics that drive it. This point is particularly relevant in the context of wireless networks, where link statistics are hard to estimate and often must be paired with simplifying assumptions (e.g., independence).

The challenge is therefore to select and prioritize relays in a distributed way, without explicitly computing this distribution.

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

We believe that this work provides a useful framework for the evaluation, analysis, and design of opportunistic routing protocols and broadens the scope of opportunistic routing in the context of low-power wireless communications. The main contributions of this paper can be summarized as follows.

- Defining a framework for anypath routing that generalizes single-path routing and unifies related work (such as [5], [8], and [9]). This framework includes the definitions of anycast link cost, remaining path cost, anypath route, and anypath route cost.
- Formulating an algorithm that provably computes the shortest (least-cost) 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.
- Introducing a novel anycast forwarding mechanism with which anypath routing can be used to improve energy efficiency and latency of low-power wireless networks.

The rest of this paper proceeds as follows. Section II defines and motivates least-cost anypath routing, Section III gives properties of least-cost anypath routes, and Section IV introduces the LCAR algorithm. Section V investigates the interplay between LCAR and different link-layer policies and protocols. Section VI shows an application of LCAR to low-power wireless networking, and Section VII evaluates performance. Finally, Section VIII surveys related work, and Section IX concludes the paper.

II. PROBLEM OUTLINE

The underlying communication primitive used by anypath routing (AR) is link-layer *anycast*, whereby a node i transmits a packet to *any* node among a set of its neighbors. We call this set the CRS of node i , denoted $J(i)$ (or J , when i can be omitted without ambiguity); it contains all the nodes that may be used as next-hop relays for packets forwarded by i toward the destination.²

With anycast transmission, a packet may travel according to a number of different paths from a source to a destination. We define an *anypath route* \mathcal{R} from a source to a destination to be a directed graph where every node (but the source) is a successor of the source, and every node (but the destination) is a prede-

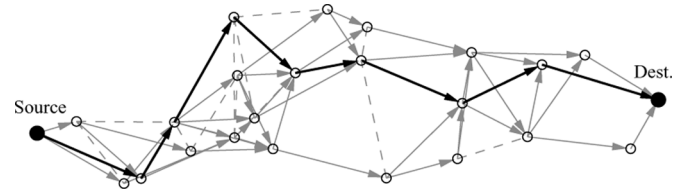


Fig. 1. Anypath route. An anypath route is the union of all possible paths from a source to a destination that are induced by a given choice of candidate relays at each node. Each node has arrows pointing to its candidate relays to reach the destination, and dashed lines to neighbors that are not candidate relays. A possible trajectory (defined in Section II-C) through the anypath route is highlighted in bold. Note that a trajectory is not always a path or walk and may contain branches, as would happen with duplicate transmissions.

cessor of the destination. An *acyclic* anypath route is an anypath route that contains no cycles. Fig. 1 shows an example of an acyclic anypath route. Each anypath route can be specified equivalently by the list of CRS $J(n_1), J(n_2), \dots, J(n_k)$ of the nodes n_1, n_2, \dots, n_k it contains or by the list of paths that can be used to traverse it. Therefore, an anypath route can also be defined as the union of all possible paths between a source and destination that arises from a given assignment of CRS at each node.

A. Why Not Use Shortest Single-Path Metrics?

Most opportunistic routing protocols are driven by single-path metrics: Nodes run a single-path routing algorithm and choose candidate relays using the shortest-path distance of their neighbors to the destination as the selection criterion. For example, a node running ExOR [5] takes as candidate relays all neighbors with lower single-path cost to reach the destination. However, single-path metrics do not accurately reflect anypath routing costs, and so can lead to suboptimal choices of candidate relays, as the following example shows.

Fig. 2 shows a network where the source has four neighbors and must select a subset of these neighbors as the set of candidate relays that it may use 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 four-node path in the mesh at the top is $p^5 = 0.24$. A single-path metric would therefore lead us to select node D as the sole candidate relay from the source. However, with anycast forwarding, each node in the upper mesh has three candidate relays to its right. A simple computation shows that the true delivery probability, when using A, B, C as candidate relays and going through the upper mesh is $(1 - (1 - p)^3)^4 \cdot p = 0.70$. If our choice of candidates is driven by single-path metrics, we 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.

B. Anycast Link Cost

We must first generalize the notion of link cost to account for anycast rather than unicast forwarding. We define the ALC $d_{i,J}$

²In the remainder of this paper, it shall be implicit when referring to a CRS that it is relative to one particular destination, which can be any node in the network.

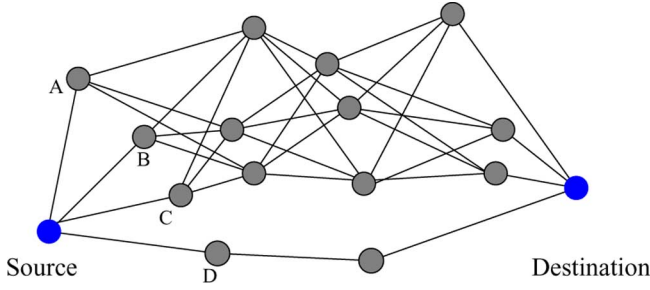


Fig. 2. Valuable detours: Why not use single-path metrics? The use of a single-path metric would prevent the source from using any of its neighbors in the upper dense area because, in single-path distance, they are farther from the destination than the source itself. However, sending a packet via the 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.

as the cost to send a packet from i to *any* node in the set J , where $J \subseteq N(i) \cup \{i\}$. Similarly to standard unicast link costs, choosing an anycast link cost is a modeling decision that depends on the cost criterion of our network. Note that for anypath routing to be worthwhile, it must be used with anycast link costs that *decrease when the candidate set is enlarged*; otherwise there is no advantage to having more than one candidate relay, and anypath routing will end up computing least-cost single-path routes.

Any ALC must have two simple properties. The first is that it reduces to the unicast link cost d_{ij} for a CRS of size 1.

Property 1: If $J = \{j\}$, then $d_{iJ} = d_{ij}$.

In other words the anycast link cost is a generalization of unicast link costs. The second property is that the ALC to any set containing the sender node is 0; this property generalizes the property $d_{ii} = 0$ of unicast distances.

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

We now give two examples of anycast link costs.

1) *Transmission-Count Metric (ETX):* As a first example of ALC, we can generalize the expected transmission count (ETX) [10] metric for unicast transmission. This metric counts the expected number of transmissions to successfully deliver a packet across an unreliable unicast link. With link-layer anycast, the ETX becomes the expected number of transmissions until any node in J receives the packet. If $i \in J$, the ETX is trivially equal to 0, while for the general case where $i \notin J$, its expression is

$$d_{iJ}^{\text{ETX}} = \frac{1}{p_{iJ}} \quad (1)$$

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

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

Of course, this definition assumes spatial independence, such that i 's transmission is received (or not) independently by nodes in J . This assumption is reasonable when fading and noise are the main source of channel errors; it may not hold when interference from other transmissions is a frequent source of errors.

Our aim here is not to derive a complex metric that captures spatial loss correlations in general conditions, but we note that the LCAR framework can accommodate such metrics and others.

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

$$d_{iJ}^{\text{E2E}} = -\log p_{iJ}. \quad (3)$$

Note that p_{iJ} increases for every node that is added to J , and so for E2E and ETX, the ALC d_{iJ} is always decreased by adding more nodes to the candidate relay set.

C. Cost of a Trajectory in an Anypath Route

A trajectory T in an anypath route \mathcal{R} is a subgraph of \mathcal{R} that connects the source and the destination. Note that a trajectory may simply be a walk (in graph language), but it can also contain branches, which could occur as a result of a duplicate transmission of a packet by more than one receiver. We now define the cost of a trajectory relative to the anypath route it traverses.

Definition 1: Let $T = (s, n_1, n_2, \dots, n_k, 1)$ 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

$$c(T|\mathcal{R}) = \sum_{i \in T} d_{iJ}(i) = d_{sJ}(s) + d_{n_1J}(n_1) + \dots + d_{n_kJ}(n_k).$$

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

D. Least-Cost Anypath Route

There are multiple possible trajectories to traverse an anypath route, and each is used with some probability $P(T)$, which depends on a number of factors such as the nondeterministic outcome of link-layer transmissions, decisions made by link- and network-layer protocol mechanisms, and the topology of the network.

Definition 2: The cost $C(\mathcal{R})$ of an anypath route \mathcal{R} is the expected cost of all trajectories across that route

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

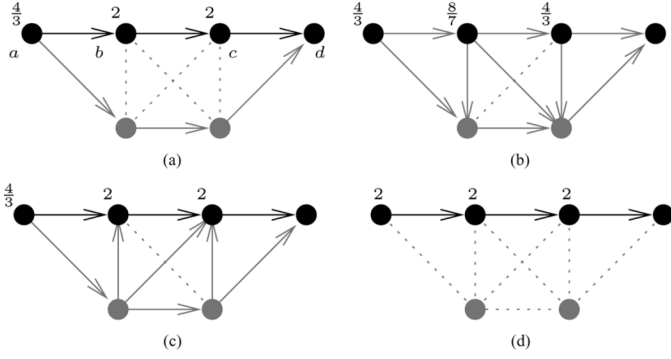


Fig. 3. Cost of a trajectory in an anypath route. Cost of the same trajectory $T = (a, b, c, d)$ traversing four different anypath routes. The cost d_{ij}^{ETX} is annotated next to nodes $a, b,$ and c . (a) $c(T|\mathcal{R}) = 5.33$. (b) $c(T|\mathcal{R}) = 3.81$. (c) $c(T|\mathcal{R}) = 5.33$. (d) $c(T|\mathcal{R}) = 6$.

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

This definition generalizes the cost of a single-path route: If all CRSs are singletons, there is only one trajectory T across an anypath route (and so $P(T) = 1$), and its cost is the sum of its constituent link costs. Each possible choice of candidate relays gives rise to a probability distribution over all possible paths between the source and destination, and this distribution determines the expected cost of using a route. Fortunately, we shall see in Section IV that it is possible to avoid the explicit computation of this distribution.

Having now defined anypath routes and their cost, it is natural to define the least-cost anypath route (“LCAR route”) between two nodes as the one with minimal cost.

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

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

III. PROPERTIES OF LEAST-COST ANYPATH ROUTES

A. Least-Cost Single-Path and Anypath Routes Can Be Disjoint

We have now seen how anypath routes generalize single-path routes and that the least-cost anypath route is equal to the least-cost single-path route when there is no gain to be had from selecting candidate relay sets of size greater than 1. It might appear natural to infer from these facts that the least-cost single-path route is always included in the least-cost anypath route. This is however not always the case, as the following example shows.

Fig. 4 shows a network with some links (solid) having delivery probability 1, some (dotted) having delivery probability $2/3$, and one link (dashed) with delivery probability $3/4$.

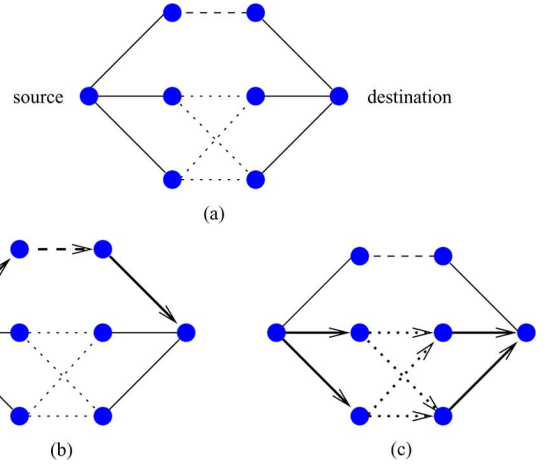


Fig. 4. Least-cost single-path and anypath routes can be disjoint. Example of a least-cost anypath route that does not include the least-cost single-path route. The cost metric is ETX. (a) Network topology: $p_{ij} = 1$ for solid links, $3/4$ for the dashed link, and $2/3$ for the dotted links. (b) Shortest single-path route. (c) Shortest anypath route.

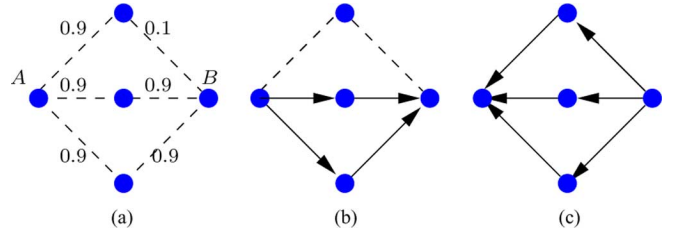


Fig. 5. Asymmetry of least-cost anypath routes. Example of a least-cost anypath route that is not symmetric. Link delivery probabilities are depicted in (a). The cost metric is ETX. (a) Network topology. (b) LCAR route from A to B . (c) LCAR route from B to A .

The link metric is ETX. Solid links therefore have cost $d_{ij} = 1$, dotted links have $d_{ij} = 3/2$, and the dashed link has $d_{ij} = 4/3$. Since $4/3 < 3/2$, the shortest single-path route will go through the top, for a total route cost of $1 + 4/3 + 1 = 10/3$. With anypath routing, the nodes on the left of the dotted links can use either of the two nodes on the right as their candidate relay set. The anycast link delivery probability is therefore $1 - (1 - 2/3)^2 = 8/9$, and so the expected transmission count of the anycast link is $9/8$, which is lower than $4/3$ for the upper (dashed) link. The least-cost anypath route thus goes through the bottom portion of the network and does not include the shortest single-path route. The least-cost anypath route has overall cost $1 + 9/8 + 1$, lower than the shortest single path cost.

B. Asymmetry of Least-Cost Anypath Routes

With single-path routes, route costs are symmetric as long as individual link costs are symmetric. This property does not hold for anypath routes. Fig. 5(a) shows a network with two endpoints 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 ETX.

Fig. 5(b) shows the least-cost anypath route from A to B . This route does not use the upper node as a candidate relay because it has a poor connection to B . Given that this upper node has

to retransmit on average 10 times to deliver a packet to B , it is preferable for node A to retransmit 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 anypath route uses all three intermediate nodes as candidate relays. Using a smaller CRS set would result in a higher ETX to get from B to the set, and since all intermediate nodes have the same delivery probability to A , there is no performance hit from using the upper relay (unlike when sending from A to B).

C. Subpaths of Least-Cost Anypaths Are Also Least-Cost Anypaths

In single-path routing, the subpaths of a shortest-path route are themselves also shortest paths. For example, if $(n_1, n_2, n_3, \dots, n_{k-1}, 1)$ is a shortest single-path route from n_1 to 1, then $(n_2, n_3, \dots, n_{k-1}, 1)$ is a shortest single-path route from n_2 to 1, $(n_3, \dots, n_{k-1}, 1)$ is a shortest single-path route from n_3 to 1, and so on. This fact is obvious and is usually not even stated explicitly in the context of single-path routing.

Does a similar property hold for least-cost anypath routes? Fortunately, the answer is positive. Before stating this formally, we must extend to anypath routes the notion of subpath used in the previous paragraph. Consider an anypath route \mathcal{R} that traverses a node i (i.e., i is not the source in this route). We say that the *sub-anypath* of \mathcal{R} from node i is the anypath route consisting of node i and all successor nodes of i in \mathcal{R} , along with their CRSs.

Proposition 1: Let \mathcal{R} be a least-cost anypath route from a source to a destination, and node k be an interior node in \mathcal{R} . Call \mathcal{R}_k the sub-anypath route of \mathcal{R} from node k , and define $D_k = \text{Cost}(\mathcal{R}_k)$. Then

$$D_k = D_k^*$$

where D_k^* is the least-cost anypath distance from node k to the destination.

This proposition (proven in the Appendix) is a crucial one, and it is fortunate, for two reasons, that it holds for least-cost anypath routes. The first is that it allows to reason recursively about routes, both in order to construct and to prove the correctness of algorithms to find these least-cost anypath routes. This is comparable to the case of single-path routing, where constructing a k -hop shortest-path route can be done starting from the shortest $(k-1)$ -hop path, recursively descending so on until $k=0$. The second reason why this proposition is helpful is that without it, each node would potentially be required to compute and keep track of one candidate relay set per (source, destination) pair, rather than having only one set per destination.

IV. FINDING LEAST-COST ANYPATH ROUTES WITH PHYSICAL COSTS

While the definition of anypath route cost (Def. 2) is intuitive, it sheds no light on how to actually *compute* this cost in a distributed setting, let alone how to find the anypath route with least cost. This section introduces a distributed algorithm to compute

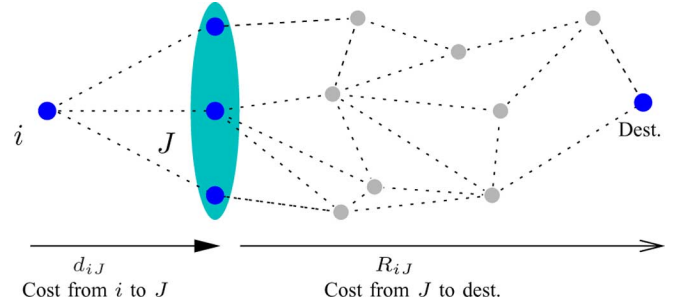


Fig. 6. Two components of anypath route cost. The cost of an anypath route can be broken down into two components: the anycast link cost, which is the cost to reach the next-hop relay, and the remaining path cost, which is the cost to get from the next-hop relay to the destination.

the optimal candidate relay sets. The LCAR algorithm is structurally similar to the classical distributed Bellman–Ford, but is driven by different metrics that generalize unicast link and path costs, respectively.

A. Remaining Path Cost

With unicast forwarding, it is trivial that the remaining cost for a packet to reach the destination after it is forwarded to the relay is the path cost from the relay to the destination. With anycast forwarding, the effective relay can be any node in J , and so the corresponding notion must be revisited. We define the RPC, denoted $R_{i,J}$, as the expected cost to reach the destination from the CRS J to which node i has anycast a packet. The breakdown of an anypath route’s cost into ALC and RPC is illustrated in Fig. 6.

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

Like for the anycast link cost, choosing the RPC is a modeling decision; it should reflect network costs as well as the underlying link-layer coordination costs. As a simple example of RPC, consider an ideal anycast link layer operating as follows. The sender i transmits a packet. If a single node in J receives the packet, that node is used as the relay. If multiple nodes in J receive the packet, then the receiver with lowest cost (the “best” receiver) to reach the destination is selected as the relay. If the packet is not received by any node in J , the sender retransmits. The behavior of nonideal, practical link layers can also be captured in the RPC and is further discussed in Section V.

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 $R_{i,J} = D$. Now consider the case where all D_k are not equal, but all link delivery probabilities are equal to some p . In this case, the RPC can be computed as

$$R_{i,J}^{\text{best}} = \frac{p}{1 - (1-p)^n} \sum_{j=1}^n (1-p)^{j-1} D_j \quad (4)$$

where it is assumed (without loss of generality) that the nodes in J are sorted by their cost to the destination, i.e., that $D_1 <$

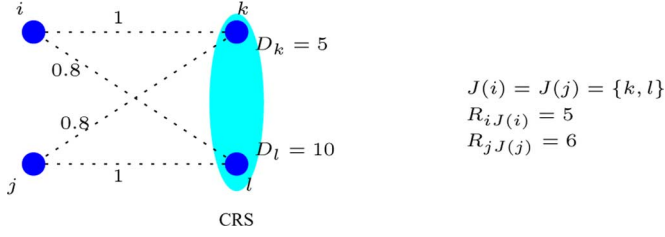


Fig. 7. Remaining path cost depends not only on the CRS J , but also on the sender i .

$D_2 < \dots < D_j$. Finally, in the general case, each node k in J receives the packet with some probability p_{ik} . The remaining path cost is then

$$R_{iJ}^{\text{best}} = \frac{1}{1 - \prod_{k \in J} p_{ik}} \left(p_{i1} D_1 + \sum_{j=2}^n p_{ij} D_j \left(\prod_{k=1}^{j-1} \overline{p_{ik}} \right) \right). \quad (5)$$

Note that like the anycast link cost, the RPC generalizes the single-path case: When $|J| = 1$, it simply becomes the cost from the relay to the destination. Note, however, that the RPC from a CRS J to the destination depends not only on J itself, but also on the predecessor node i of J . In other words, the same CRS J can give a different RPC for two different senders i . In this sense, RPC is quite different to the single-path notion that it generalizes since the traditional single-path notion of subpath cost does not depend on the previous node in a path. This dependence of the RPC on i is only through the distribution on J . For a given destination, if i and i' induce the same distribution over J , then the RPCs are equal.

We illustrate the dependence of the RPC on the sender with the example of Fig. 7. Two forwarding nodes i and j each have the same candidate relay set $C = J(i) = J(j) = \{k, l\}$. We consider the ideal link layer described earlier, which selects as relay the receiver with lowest cost to reach the destination. Thus, k is chosen as relay every time it receives a packet (because $D_k < D_l$), and node l is only chosen if it receives a packet and k does not. What are the remaining path costs R_{iC} and R_{jC} to the destination? Consider first sender i . Node k receives every packet from i , and so it is the effective relay for every packet (including for packets that node l had received as well). The remaining path cost R_{iC} is thus 5. The situation is different for sender j . Node k only receives packets from j with probability 0.8, whereas node l receives every packet, thus $R_{jC} = 0.8 * 5 + 0.2 * 10 = 6$. Note that an ideal link layer is not required to illustrate $R_{iC} \neq R_{jC}$, e.g., the inequality would also hold if the relay is selected at random each time both k and l receive a transmission.

B. Physical Cost Models

The ALC and RPC metrics can be designed in many different ways depending on the underlying protocol and cost model. However, they must jointly satisfy one criterion in order for routing to converge. It is called the physical cost criterion and can be interpreted as a generalization of the single-path routing requirement that link costs be nonnegative in order for routing to converge. The criterion requires that if a node i adds to its CRS

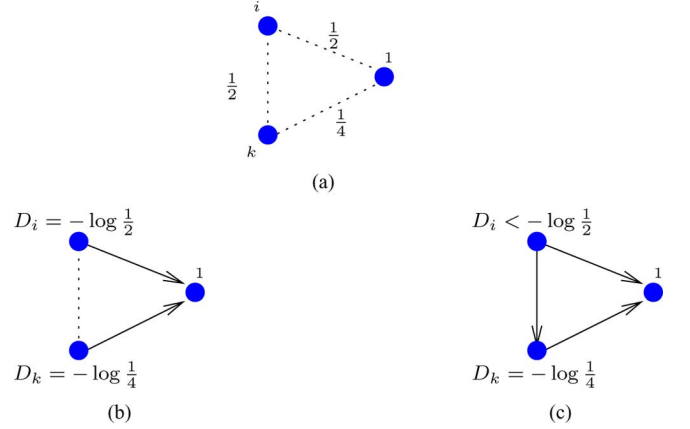


Fig. 8. E2E anycast link cost is not a physical cost model (counterexample for proof of Property 4, discussed in Section IV-B). In this example, the cost from i to the destination *decreases* if it adds as candidate relay a node k , which in (b) has *higher cost* than i itself has to reach the destination. (a) Network with source i and destination 0. (b) $D_i < D_k$. (c) D_i decreases.

a neighbor with higher cost to the destination than i itself, then i 's cost to reach the destination must increase, even though the anycast link cost d_{iJ} may have decreased by adding this node.

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

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

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

We now return to the two the ALCs and RPCs defined in Section II-B.

Property 3: The ETX anycast link cost (1) is a physical cost model.

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

We leave the proof of Property 3 as an exercise to the reader. We prove that the E2E metric is not physical by using a counterexample. Consider the network of Fig. 8(a), where node 1 is the destination, and where delivery probabilities are annotated next to each link. In Fig. 8(b), node i has as candidate relay set the singleton set containing the destination only, i.e., $J = \{1\}$. The remaining path cost is therefore $R_{iJ} = 1$, and we have $D_i = d_{iJ} + R_{iJ} = -\log 1/2$. Node k has the same candidate relay set consisting of the destination only, giving us $D_k = -\log 1/4 > D_i$.

Consider now Fig. 8(c), where node i uses as CRS the set $J' = \{1, k\}$. If the delivery probability ALC were a physical cost model, the cost at node i would increase with this candidate relay set J' because $D_k > D_i$. More precisely, we should have $d_{iJ'} + R_{iJ'} \geq d_{iJ} + R_{iJ}$, where $J = \{1\}$. However, we can compute that

$$\begin{aligned} d_{iJ'} &= -\log(1 - (1 - p_{i1})(1 - p_{ik})) \\ &= -\log \frac{3}{4} \end{aligned}$$

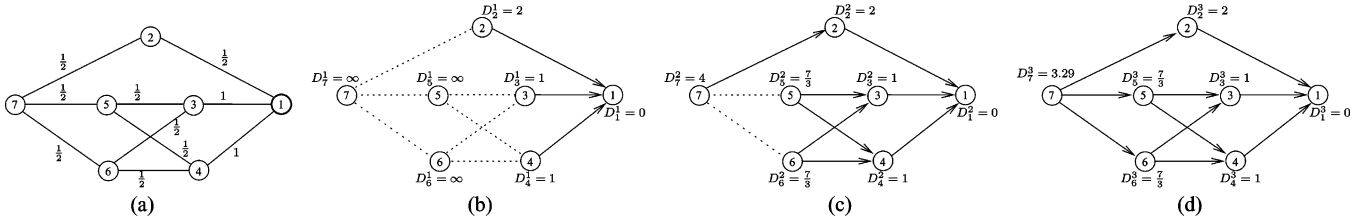


Fig. 9. Illustration of the LCAR algorithm computing least-cost anypath routes from every node to the destination. (a) Initial graph with link probabilities annotated. The destination is node 1. (b) $h = 1$: least-cost (≤ 1) anypath routes. (c) $h = 2$: least-cost (≤ 2) anypath routes. (d) $h = 3$: least-cost (≤ 3) anypath routes.

and

$$\begin{aligned} R_{iJ} &= -\log((1 - (1 - p_{i1})(1 - p_{ik})) \\ &\quad \times (p_{i1} + (1 - p_{i1})p_{ik}p_{k1})) \\ &= -\log \frac{3}{4}. \end{aligned}$$

Putting both together, we obtain

$$d_{iJ} + R_{iJ} = -\log \frac{3}{4} - \log \frac{3}{4} < -\log \frac{1}{2} = d_{iJ} + R_{iJ}$$

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

C. Least-Cost Anypath Routes Are Acyclic With Physical Costs

Our definition of anypath routes allows the presence of cycles. Is this necessary, and in particular, can the least-cost anypath route ever contain a cycle?

Proposition 2: In a physical cost model, no cyclic anypath route has lower cost than the least-cost 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 least-cost 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, the following somewhat counterintuitive point: In a nonphysical cost model, the least-cost anypath route *may contain cycles*. In the case of the nonphysical E2E metric, one way to interpret why the least-cost 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 best 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. This is somewhat reminiscent of hot-potato routing [11] for wired networks with limited buffer sizes, where it is preferable to forward a packet to a node that is farther to the destination than to simply drop it, in the case that the link to the next-hop node is busy.

D. Finding Least-Cost Anypath Routes With Physical Costs

We now show an algorithm that computes least-cost anypath routes. We assume throughout this section the use of a physical

cost model; therefore, we have from Proposition 2 that the least-cost anypath routes in this section are acyclic.

How does a node select which of its neighbors should be candidate relay nodes? As illustrated in Fig. 6, the expression to minimize is the sum of the ALC and RPC, which must be minimized over all possible subsets $J \subseteq N(i)$

$$D_i = \min_{J \in 2^{N(i)}} [d_{iJ} + R_{iJ}]. \quad (6)$$

We call this “the anypath Bellman equation.” It represents the steady state of the LCAR algorithm, which computes least-cost anypath 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 anypath routing cost estimate from i to the destination at the h th iteration; it converges toward D_i . By convention, we take

$$D_1^h = 0, \quad \text{for all } h \quad (7)$$

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

$$D_i^{h+1} = \min_{J \in 2^{N(i)}} [d_{iJ} + R_{iJ}^h], \quad \text{for all } i \neq 1 \quad (8)$$

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

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

The algorithm terminates when

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

Just like single-path Bellman–Ford, the LCAR algorithm can work in a distributed setting, with nodes asynchronously recomputing their cost [using (8)] and advertising it to their neighbors. An example is shown in Fig. 9.

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

Proposition 3: The LCAR algorithm computes, at iteration h , the least-cost ($\leq h$) anypath 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 anypath route from i to the destination.

The proof of this proposition is given in the Appendix.

While the upper bound on the LCAR algorithm's convergence time (in number of iterations) is the same as for single-path Bellman–Ford, its complexity is greater since there are $2^{|N(i)|}$ possible subsets that must be evaluated, compared to $|N(i)|$ possible relays with single-path routing. Cases where the complexity of computing (8) can be reduced are discussed in [1]; under certain criterion of practical relevance on the ALC and RPC, computing (8) requires only searching through n possible sets.

V. RELAY SELECTION POLICIES AND LINK-LAYER COORDINATION PROTOCOLS

In the previous section, we assumed an ideal link-layer anycast that had no overhead, always chose the best effective receiver as next-hop relay, and never let through duplicate transmissions. This section considers other, nonideal link-layer anycast mechanisms. It uses LCAR to investigate the interplay between least-cost anypath routes and the underlying link-layer anycast coordination protocol. It shows how LCAR can take into account anycast selection policies that (intentionally or unintentionally) might not select the best receiver as the next hop relay. It also shows how LCAR can take into account the link-layer coordination cost of anycast forwarding and the possibility of erroneous duplicate transmissions.

A. Other Policies for Effective Relay Selection

When a packet transmitted by a node i is received by more than one node in i 's CRS, a decision must be made as to which receiver should then forward the packet further. We call this an effective relay selection (ERS) policy. It may appear that the ERS policy is only relevant to the link layer since it is carried out on a per-packet basis as part of anycast forwarding. However, it is also directly relevant to the network layer because a routing decision that selects optimal candidate relay sets must take into account the ERS policy that is used. In other words, the least-cost anypath route using one ERS policy is not the same for another ERS policy; this is incorporated into the LCAR algorithm via the distribution over J that is reflected in the remaining path cost.

The policy described in Section IV-A always chooses as next forwarding node the “best-placed” receiver, that is, the receiver k with minimum cost to the destination D_k . We call this policy *ERS-best*. Another example of ERS policy is *ERS-any*, where the relay is chosen uniformly at random among receivers of a packet; we show here how *ERS-any* can be modeled in the LCAR framework.

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

$$R_{iJ}^{\text{any}} = \sum_{S \subseteq 2^{\mathcal{J}(i)}} P(S) \left(\frac{1}{|S|} \sum_{j \in S} D_j \right) \quad (9)$$

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

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

By plugging the above expression of R_{iJ}^{any} into equation (8), we obtain a different instance of LCAR that computes the least-cost routes under the use of *ERS-any*. Note that not only the *costs* of routes will be different with *ERS-any* than *ERS-best*, but the anypath routes themselves will in the general case be different because the J minimizing (8) may not be the same under different expressions of R_{iJ} . Intuitively, with *ERS-any*, a neighbor with a high D_k that is added to the CRS is more likely to be used than with *ERS-best*, and so the optimal CRS with *ERS-any* tends to be smaller than with *ERS-best*.

By definition, *ERS-best* makes optimal choices. So why should one consider *ERS-any*, or any other ERS policy? Because the cost and complexity of executing an anycast link-layer coordination protocol may vary depending on the ERS policy. Even if an ERS policy sometimes selects suboptimal relays, the overall cost of using it cannot be a priori ruled to be higher than *ERS-best*. For example, *ERS-best* has stricter selection requirements than *ERS-any*, and so a link-layer protocol implementing it will likely have higher overhead. It is therefore useful to be able to model other ERS policies, both in order to evaluate their impact on routing costs and to build protocols that may incorporate these policies.

ERS-best and *ERS-any* are only two simple examples of possible ERS policies. One could imagine many other ERS policies—for example, a hybrid between *ERS-any* and *ERS-best*, which would select any receiver among the k best, thus operating in an intermediate regime between the hard coordination constraints of *ERS-best* and the suboptimal relay choice of *ERS-any*. Or, one could design an ERS policy that takes $|N(i)|$ (or J) into account, for example making the value k of the hybrid policy described depend on the neighborhood density. An in-depth investigation of the different ERS policies and their link-layer implementations is beyond the scope of this paper; however, we note that the LCAR algorithm can accommodate and find optimal routes for any ERS, as long as the resulting distribution over J is correctly modeled in the RPC.

B. Duplicate Relays

An important challenge in opportunistic and anypath routing is the design of a coordination protocol to implement an ERS policy. This protocol must ensure that the nodes receiving a packet all agree and select the correct relay in a distributed way. While an ideal protocol does this 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, because of lost signalling information, when two nodes mistakenly believe they are each the only receiver of a packet.

The LCAR framework and algorithms also can capture and account for such imperfections in coordination protocols. For example, consider an implementation of *ERS-any* where each node in J , other than the effective relay, mistakenly forwards a

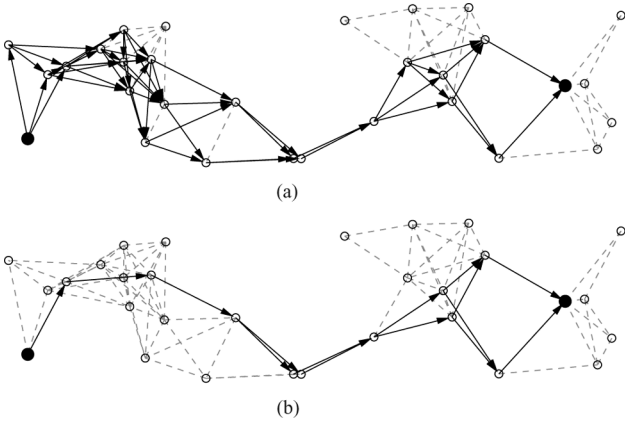


Fig. 10. Least-cost anypath routes with perfect versus imperfect link layers. Comparison of least-cost anypath routes with perfect link layer coordination versus a link layer that sometimes lets through duplicate transmissions. In (b), the LCAR route has smaller CRSs because the reduction in forwarding cost from using large CRS is offset by the cost of possible duplicate transmissions. This effect is particularly pronounced at large distances from the destination. (a) R^{any} . (b) R^{dup} .

duplicate packet with probability p . In such a case, the RPC can be expressed as

$$R_{i,J}^{\text{dup}} = (1 + p(|J| - 1)) \cdot R_{i,J}^{\text{any}}. \quad (10)$$

This R^{dup} can then be used as the RPC in the LCAR algorithm in order to obtain anypath routes that take into account the expected cost of duplicates. The result is that sizes of CRS in an LCAR route are smaller when using as RPC R^{dup} than R^{any} because the possibility of duplicates increases the cost of having large CRSs, which can partially (or entirely) offset the reduced forwarding cost captured by the ALC. In fact, in the extreme case (e.g., with high duplication probability and with an ALC metric that decreases “slowly” with $|J|$), LCAR could end up using only CRSs of size 1, in effect computing shortest-path routes.

A comparison of LCAR routes found by the algorithm using R^{dup} versus R^{any} is shown in Fig. 10. These routes are computed via simulation, using a high value of $p = 0.1$, in order to make the differences strongly apparent. An interesting observation is that the degree to which R^{dup} causes the LCAR algorithm to “clamp down” on CRS sizes depends on the distance to the destination.

At close distance to the destination (e.g., one or two 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 small number of hops. If a duplicate is generated earlier on in the route, however, it will potentially travel more hops throughout the network, resulting in more wasteful transmissions; thus, the least-cost anypath route with duplicates has smaller CRS sizes close to the source and larger CRS sizes close to the destination.

VI. APPLICATION TO LOW-POWER WIRELESS NETWORKS

Just like Bellman–Ford can be used with a wide variety of cost and distance metrics, so can LCAR. This section shows how LCAR 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 [12], [13], it is necessary to power it down whenever possible by using some form of duty cycling of the radio. 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) [12], 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 LCAR. Note that LCAR can also be used with other low-power link schemes; we illustrate it with LPL because of LPL’s widespread adoption in a large number of wireless sensing projects due to its simplicity and robustness.

A. 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.

Note that in practice the duration of the listen state is lower-bounded by a minimal wakeup time, and hence decreasing ρ requires to increase t_{rx} . 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 known 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.

B. Anycast Low-Power Listening (A-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 that 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.

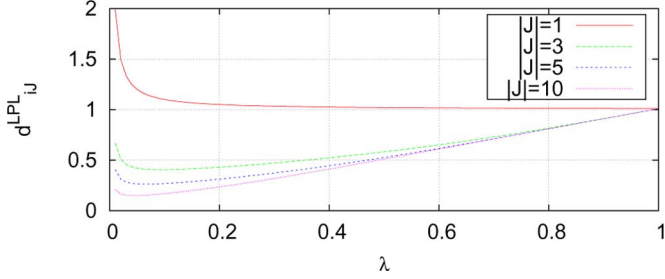


Fig. 11. Anycast LPL link costs. Anycast link cost $d_{i,J}^{A-LPL}$ as a function of preamble length λ for different CRS sizes. In this figure, we have set $t_{\text{pkt}} = 0.01 \cdot t_{\text{rx}}$, in line with existing implementations of LPL [12], [13]. With a CRS of size 3, transmission cost is reduced by a factor of 2.5 over unicast LPL; with a CRS of size 10, it is reduced by a factor greater than 5.

Assume that wakeup times are independent and uniformly distributed within the interval t_{rx} . Assume that we use a preamble of length $t_{\text{pre}} < t_{\text{rx}}$. Define $\lambda = t_{\text{pre}}/t_{\text{rx}}$, 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 CRS of size n is

$$p_{\text{hit}} = 1 - (1-\lambda)^n. \quad (11)$$

While p_{hit} increases with the size of the CRS, guaranteeing that some node in the CRS receives a packet (e.g., reaching $p_{\text{hit}} = 1$) 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 retransmission strategy. The average number of transmissions until we hit at least one node will be $1/p_{\text{hit}}$.

Anycast link cost metric for A-LPL must reflect an entire transmission cost, which includes both preamble and packet transmission time.³ Note the tradeoff between decreasing λ (cost of a single transmission) and increasing $1/p_{\text{hit}}$ (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 A-LPL

$$d_{i,J}^{A-LPL} = \min_{t_{\text{pre}} \in [0, t_{\text{rx}}]} \frac{t_{\text{pre}} + t_{\text{pkt}}}{p_{\text{hit}}} = \min_{\lambda \in [0, 1]} \frac{\lambda t_{\text{rx}} + t_{\text{pkt}}}{1 - (1-\lambda)^{|J|}} \quad (12)$$

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 $d_{i,J}^{A-LPL} = t_{\text{pre}} + t_{\text{pkt}}$, which is equal to the forwarding cost in the unicast case. Computing $d_{i,J}^{A-LPL}$ analytically is hard because minimizing (12) requires finding the zeroes of an order- $|J|$ polynomial. We therefore compute it numerically and plot it in Fig. 11. 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 CRS sizes ($|J| \leq 10$).

³Energy is proportional to transmission time under the assumption of fixed transmit power.

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 (11). Thus, the remaining path cost is obtained by substituting λ_{opt} for p in (5), where λ_{opt} is the argument minimizing (12).

C. Link-Layer Coordination With A-LPL

Having an efficient and robust link-layer coordination protocol is a key challenge with anycast forwarding. With A-LPL, however, the burden on a candidate coordination protocol is much smaller than with a non-duty-cycled link layer such as [5]. This is because it is rare that multiple nodes receive the same packet, due to their radios being turned on only a very small fraction of the time (and under our independence assumption). Fig. 11 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 our protocol implementation of A-LPL (described in [1]), each node in a CRS that receives a packet forwards it; we found that a coordination protocol was not worth its cost given the minute number of duplicates that actually happened.

VII. PERFORMANCE

This section evaluates the performance of LCAR in comparison to standard single-path routing and with anypath routing using single-path metrics. This evaluation uses simulations with a simple network and channel model and focuses on low-power routing with LPL. Due to lack of space, it does not cover throughput performance under ETX and does not describe our implementation and evaluation on a 50-node wireless test bed; these results are available in [1].

We use the following terminology: *SP* routes are least-cost single-path routes as found by classical Bellman–Ford or Dijkstra algorithms. *LCAR* routes are the least-cost anypath routes found by the algorithm of Section IV. Finally *SP-AR* routes are anypath routes obtained using a single-path metric (such as in ExOR) as discussed in Section II-A.

A. Route Costs

We first evaluate the cost of paths found by LCAR. 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 $\pm 10\%$ of the empirical mean.

The first set of simulations evaluates the cost of LCAR and SP-AR routes. The unit of route cost is transmission *duration*, counting both preambles and packets. For each node pair, we compute the shortest (least-cost) single-path route and the LCAR route using Bellman–Ford and the LCAR algorithm. In the LCAR case, we compute the anycast link cost and remaining path cost as defined Section VI-B. We then order nodes by

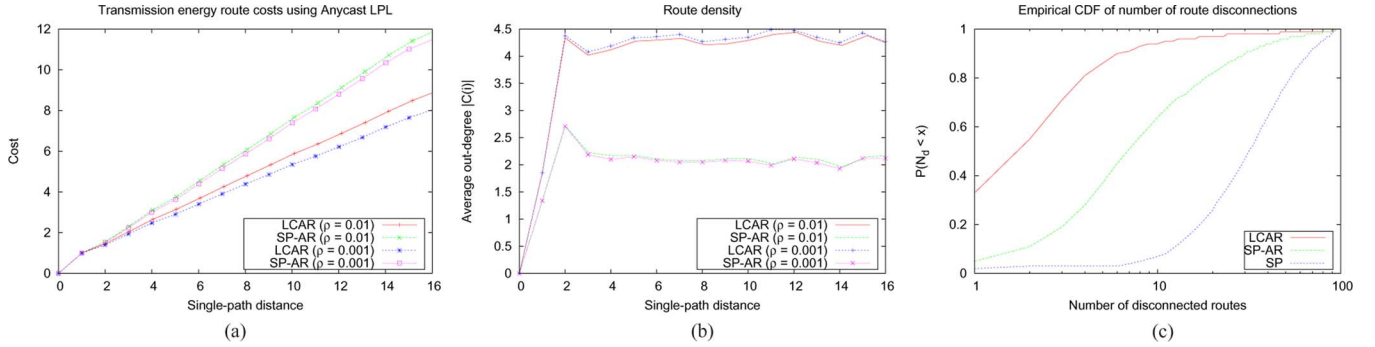


Fig. 12. Simulation results for a 500-node network and average density 10. (a) Comparison of average LCAR costs using ALC metric $d_{ij}^{\text{A-LPL}}$ as a function of shortest single-path distance, and for two different cycles. (b) Average out-degree $|J(i)|$ as a function of single-path distance to the destination. (c) Empirical CDF of N_d , the number of disconnected routes, for the link addition/deletion model with $p_v = 0.05$.

single-path distance and, in Fig. 12(a), plot the average LCAR and SP-AR costs as a function of this shortest single-path distance. The cost of the SP-AR routes is approximately 40% higher than that of LCAR routes. 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 (12) can use lower values of λ as t_{rx} is increased. Note that LCAR route costs are (roughly) a constant factor of single-path costs; the cost reduction of LCAR routes increases with *density* rather than *diameter*.

One way to characterize an anypath route is to consider the number of candidate relays that nodes in this route have. Do LCAR routes have more candidate relays than SP-AR routes? Or is the lower cost of LCAR due to the choice of CRS being informed by the more suitable ALC metric, but with similar CRS 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 Fig. 12(b), averaged over 10 000 network realizations. This shows that LCAR routes are able to use more candidate relays than SP-AR routes; nodes in LCAR routes have about four candidates, in comparison with two for SP-AR routes. An example using a simulated network of 20 nodes is given in Fig. 13.

B. 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 anypath 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 LCAR or SP-AR routes, disconnectedness means that the route has no possible trajectory to reach the destination.

We plot the empirical CDF of N_d in Fig. 12(c). As expected, LCAR routes have fewer disconnections than SP. More interesting, however, is that LCAR routes are also significantly less

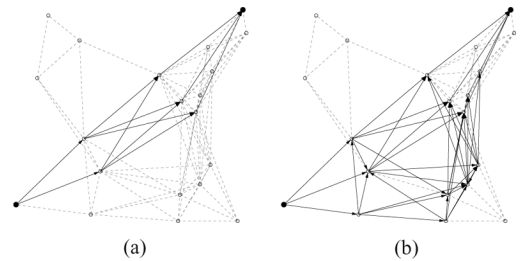


Fig. 13. Comparison of SP-AR and LCAR routes in a simulated network. The source is in the bottom left corner, and the destination is in the top right corner. (a) With SP-AR, nodes can only take candidate relays that are closer in *single-path* distance. (b) With LCAR, the distance metric takes into account the true cost of anycast forwarding, and so allows to use more candidate relays at each node. As a result, the cost of the LCAR route is lower.

prone to disconnection than SP-AR routes. For example, the probability that *less* than 10% of routes are disconnected with LCAR is over 0.95, while it is only 0.65 with SP-AR. This is a direct consequence of LCAR's larger CRS sets, as shown in Fig. 12(b). Note that robustness is a multifaceted 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 LCAR cost metric provides routes that are more stable to such noisy inputs than SP or SP-AR; measurements confirming this intuition are given in [1].

VIII. RELATION TO EXISTING WORK

Link-layer anycasting has been previously proposed and motivated in various forms [3], [7]. These works focus on *mechanisms* to implement anycast forwarding at the link layer, and assume that the network layer maintains a list of possible relay candidates (e.g., by a multipath 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 LCAR algorithm could be used to feed these link layers with relay candidates.

Jain and Das [6] go a step further by integrating an anycast extension of the IEEE 802.11 link layer with the multipath AODV (AOMDV) [14] routing protocol. They observe the same tradeoff as [7] 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 to improve load balancing, redundancy, or failover by providing multiple route choices. This is in contrast with LCAR (and opportunistic routing in general), which provides multiple relay candidates specifically to take advantage of anycast forwarding. One example of such multipath routing in a wireless network is the work of Srinivas and Modiano [15], 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, one example of multipath routing is the work of Zaumen and Garcia-Luna-Aceves [16]. 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 *least-cost* anypath 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 [17], and select as candidate relays those nodes that are closer to the destination than the current node. This approach is simple and trivially guarantees loop freedom. However, one challenge inherent in a geographic approach is avoiding dead ends. Another is that radio propagation is highly irregular at local scales, and so making progress in physical distance does not guarantee making progress in the actual network topology.

Our 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 [12], [18], [19]. They do not modify these link protocols however to specifically take advantage of anycast forwarding.

More recently, the use of opportunistic routing with multirate transmission has been studied. Radunovic *et al.* [20] introduce an optimization framework that is used jointly for rate adaptation, scheduling, and routing. Zeng *et al.* [21] investigate opportunistic routing with multiple rates and take into account transmission conflicts. In both cases, the problem is NP-hard, and heuristics are used to find a solution. Laufer *et al.* [22] build upon the same work as does this paper [1], [2], and propose a generalization of Dijkstra's algorithm for multirate opportunistic routing. The intersection of network coding and opportunistic routing is another promising area of investigation; recent work in this area includes [23] and [24].

Lott and Teneketzis independently propose opportunistic forwarding in [25]. They formulate the routing problem under the assumption that the global state is known, which is the set of nodes in the network that have received the packet so far. The optimal Markov policy can be computed through dynamic programming, exploiting the special structure of the problem. Although they give distributed algorithms to compute the Markov policy itself, the forwarding decisions under this policy are functions of the global state. This would incur a significant control

overhead in a protocol implementation. In contrast, LCAR only requires local coordination for forwarding decisions once the CRSs have been computed. Another formal contribution in the area of opportunistic forwarding is the work of Lu *et al.* [26], who designed a routing algebra for opportunistic routing.

Finally, the work of Zhong *et al.* and Chachulski [8] are most closely related to this paper. Zhong *et al.* [9] study the specific case of ETX-based transmission and come up with an expression similar to the RPC used in this paper. They also propose an algorithm for selecting and prioritizing candidate relays. This algorithm is specific to the ETX case and is not proven to be optimal. Chachulski [8] also provides an algorithm (based on Dijkstra's algorithm) to compute candidate relay sets. Like Zhong *et al.*'s, the algorithm is specific to ETX-based transmission, and the author does not provide a proof of optimality.

IX. CONCLUSION

This paper introduces an algorithm to compute least-cost anypath routes in multihop wireless networks. The technique is general, and the associated framework can accommodate a number of different network and cost models. We believe that LCAR can be useful not only as an algorithmic building block for implementing anypath routing protocols, but also as a protocol-modeling framework to investigate design questions such as the tradeoff between simplifying link-layer anycast coordination mechanisms and having higher routing costs.

APPENDIX

Proof of Proposition 1: Call \mathcal{T} the least-cost anypath route from node k to the destination. We have therefore $\text{Cost}(\mathcal{T}) = D_k^*$. Since \mathcal{T} is the least-cost anypath from k to the destination, we cannot have $D_k^* > D_k$, or otherwise \mathcal{R}_k would be a shorter anypath 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 anypath route \mathcal{R} , of which \mathcal{R}_k is a sub-anypath. 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 least-cost anypath route.

Proof of Proposition 2: By contradiction, assume that the least-cost anypath route \mathcal{R} between two nodes is not acyclic. It therefore contains at least one cycle. Let us consider one cycle $(n_1, n_2, n_3, \dots, n_{k-1}, n_1)$, with nodes having destination distances $(D_1, D_2, D_3, \dots, D_{k-1}, D_1)$. Since this is a cycle, one of two cases must be true: Either $D_1 = D_2 = D_3 = \dots = D_{k-1}$, 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:** $D_1 = D_2 = D_3 = \dots = D_{k-1}$. 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 $(n_1, n_2, n_3, \dots, n_{k-1}, n_1)$ 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 $J(n_i) = \{n_{i+1}\} \cup K$.) Now, if we consider the definition (Definition 4) 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 $J(i) = K$) 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 $D_i = d_{i\{j\}} + R_{i\{j\}} = d_{i\{j\}} + D_j > D_j$). Again, considering the definition (Definition 4) 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 least-cost anypath route \mathcal{R} , let us construct the anypath \mathcal{R}' by applying either of the two modifications above to every cycle in \mathcal{R} . Calling D'_k the distance from node k to the destination, we have that $D'_k \leq D_k$ for all nodes, with equality for at least one node (the one whose candidate relay set was pruned according to the modifications above).

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

Proof of Proposition 3: We prove the first part of the proposition by induction over h .

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

$$D_i^1 = d_{i1}, \quad \text{for all } i \neq 1$$

which is indeed the least-cost (≤ 1) anypath distance to the destination.

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

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

This route consists of $|J(i)|$ links from i to each node in its CRS $J(i)$, and then of $|J(i)|$ sub-anypath routes from each node in $J(i)$ to 1 that each have a h -hop longest trajectory. From Proposition 1, we know that these sub-anypath routes must be least-cost 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 ($\leq h$) trajectory

$$\begin{aligned} \text{Cost}(\mathcal{R}_i^{h+1}) &= d_{iJ(i)} + R_{iJ(i)} \\ &= \min_{J \in 2^{N(i)}} [d_{iJ} + R_{iJ}^h] \\ &= D_i^{h+1}. \end{aligned}$$

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

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

and so D_i^{h+1} is the least-cost 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.

REFERENCES

- [1] H. Dubois-Ferrière, "Anypath routing" Ph.D. dissertation #3636, School of Comput. & Commun. Sci., EPFL, Lausanne, Switzerland, 2006.
- [2] H. Dubois-Ferrière, M. Grossglauber, and M. Vetterli, "Least-cost opportunistic routing," in *Proc. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, Sep. 26–28, 2007, pp. 994–1001.
- [3] P. Larsson, "Selection diversity forwarding in a multihop packet radio network with fading channel and capture," *SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 4, pp. 47–54, 2001.
- [4] B. Zhao and M. C. Valenti, "Practical relay networks: A generalization of Hybrid-ARQ," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 7–18, Jan. 2005.
- [5] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," in *Proc. ACM SIGCOMM*, Philadelphia, PA, 2005, pp. 133–144.
- [6] S. Jain and S. R. Das, "Exploiting path diversity in the link layer in wireless ad hoc networks," in *Proc. IEEE WoWMoM*, 2005, pp. 22–30.
- [7] 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.
- [8] S. Chachulski, "Trading structure for randomness in wireless opportunistic routing," Master's thesis, Dept. Comput. Sci., MIT, Cambridge, MA, 2007.
- [9] Z. Zhong and S. Nelakuditi, "On the efficacy of opportunistic routing," in *Proc. 4th IEEE SECON*, San Diego, CA, 2007, pp. 441–450.
- [10] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. 9th ACM MobiCom*, San Diego, CA, 2003, pp. 134–146.
- [11] A. Greenberg and B. Hajek, "Deflection routing in hypercube networks," *IEEE Trans. Commun.*, vol. 40, no. 6, pp. 1070–1081, Jun. 1992.
- [12] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM SenSys*, Los Angeles, CA, Apr. 2004, pp. 95–107.
- [13] H. Dubois-Ferrière, R. Meier, L. Fabre, and P. Metrailler, "TinyNode: A comprehensive platform for wireless sensor network applications," in *Proc. 5th IPSN*, Nashville, TN, Apr. 2006, pp. 358–365.
- [14] M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proc. 9th ICNP*, 2001, pp. 14–23.
- [15] A. Srinivas and E. Modiano, "Minimum energy disjoint path routing in wireless ad-hoc networks," in *Proc. 9th Annu. ACM MobiCom*, New York, 2003, pp. 122–133.
- [16] W. T. Zaumen and J. J. Garcia-Luna-Aceves, "Loop-free multipath routing using generalized diffusing computations," in *Proc. IEEE INFOCOM*, 1998, pp. 1408–1417.
- [17] R. R. M. Zorzi, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance," *IEEE Trans. Mobile Comput.*, vol. 2, no. 4, pp. 337–348, Oct.–Dec. 2003.
- [18] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. 1st ACM SenSys*, New York, 2003, pp. 171–180.
- [19] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, pp. 1567–1576.

- [20] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez, "An optimization framework for opportunistic multipath routing in wireless mesh network," in *Proc. IEEE INFOCOM*, Phoenix, AZ, 2008, pp. 2252–2260.
- [21] K. Zeng, W. Lou, and H. Zhai, "Capacity of opportunistic routing in multi-rate and multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pt. 1, pp. 5118–5128, Dec. 2008.
- [22] R. Laufer, H. Dubois-Ferrière, and L. Kleinrock, "Multirate anypath routing in wireless mesh networks," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, 2009, pp. 37–45.
- [23] S. K. S. Chachulski, M. Jennings, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM SIGCOMM*, Kyoto, Japan, 2007, pp. 169–180.
- [24] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, 2008, pp. 361–365.
- [25] C. Lott and D. Teneketzis, "Stochastic routing in ad-hoc networks," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 52–70, Jan. 2006.
- [26] M. Lu and J. Wu, "Opportunistic routing algebra and its applications," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, 2009, pp. 2374–2382.



Henri Dubois-Ferrière received the Diplôme d'Ingénieur en Systèmes de Communication and the Ph.D. degree in 2006, both from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2000 and 2006, respectively.

During his undergraduate and graduate studies, he spent time at Carnegie Mellon University, Pittsburgh, PA; Eurecom Institute, Sophia-Antipolis, France; and the University of California, Los Angeles. From 2000 to 2002, he worked at FastForward Networks and Inktomi, both in San Francisco, CA. He is

currently with Riverbed Technology, which he joined in 2007.



Matthias Grossglauser received the Diplôme d'Ingénieur en Systèmes de Communication from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 1994; the M.Sc. degree from the Georgia Institute of Technology, Atlanta, in 1994; and the Ph.D. degree from the University Pierre et Marie Curie (Paris 6), Paris, France, in 1998.

He is with the Nokia Research Center (NRC), Helsinki, Finland, currently as the head of the Data Insight program, an interdisciplinary team based in

Helsinki and Palo Alto, CA, focused on data analytics, machine learning, and visualization. He is also a member of Nokia's CEO Technology Council. He is affiliated with the School of Computer and Communication Sciences (I&C), EPFL, as an Assistant Professor. From 2007 to 2008, he was the Director of the Internet Laboratory at NRC. From 1998 to 2002, he was a member of the Networking and Distributed Systems Laboratory at AT&T Labs-Research, Florham Park, NJ. His research interests are in mobile and wireless networking, social and complex networks, and network traffic measurement and modeling.



Martin Vetterli (S'86-M'86-SM'90-F'95) received the Dipl. El.-Ing. degree from ETH Zurich (ETHZ), Zurich, Switzerland, in 1981; the M.S. degree from Stanford University, Stanford, CA, in 1982; and the Doctorat ès Sciences degree from Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 1986.

He was a Research Assistant with Stanford University and the EPFL and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University, New York, NY, where he was last an Associate Professor of electrical engineering and Co-Director of the Image and Advanced Television Laboratory. In 1993, he joined the University of California, Berkeley, where he was a Professor in the Department of Electrical Engineering and Computer Sciences until 1997; he now holds an Adjunct Professor position. Since 1995, he has been a Professor of communication systems at the EPFL, where he chaired the Communications Systems Division (1996–1997) and heads the Audiovisual Communications Laboratory. From 2001 to 2004, he directed the National Competence Center in Research, Switzerland, on mobile information and communication systems. He has also been a Vice-President for Institutional Affairs with EPFL since October 2004. He has held visiting positions at ETHZ and Stanford University in 1990 and 1998, respectively. His research interests include sampling, wavelets, multirate signal processing, computational complexity, signal processing for communications, digital image/video processing, joint source/channel coding, and signal processing for sensor networks.

He is a Fellow of the Association for Computing Machinery (ACM) and a Member of SIAM. He is on the Editorial Boards of *Applied and Computational Harmonic Analysis*, the *Journal of Fourier Analysis and Application*, and the *IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING*.