

# Opportunistic Routing Algebra and its Applications

Mingming Lu

Citrix System, Inc.

Fort Lauderdale, Florida 33309

Email: Mingming.Lu@citrix.com

Jie Wu

Department of Computer Science and Engineering

Florida Atlantic University

Email: jie@cse.fau.edu

**Abstract**—Opportunistic routing (OR) has received much attention as a new routing paradigm due to its efficient utilization of broadcasting and spacial diversity of the wireless medium. Although numerous OR algorithms and protocols have been proposed to apply to various environments and integrate with numerous techniques, as far as we know, none of the existing works have used mathematical tools such as routing algebra to analyze the compatibility of routing metrics and routing protocols so as to provide a guideline for routing protocol design. In this paper, we design a new OR algebra based on the routing algebra proposed for inter-domain routing [1], identify the essential properties of OR in the mathematical language of the OR algebra, and analyze the design space in terms of routing metrics for various routing requirements.

**Index Terms**—Consistency, loop-free, multi-hop wireless networks, opportunistic routing (OR), optimality, routing algebra, routing metrics.

## I. INTRODUCTION

Over the past decade, multi-hop wireless networks have received an increased amount of attention due to their broad applications and the easy deployment at low cost without relying on the existing infrastructure. However, this infrastructure-less property and the unstable nature of the wireless medium incur the problem of unreliable communication. To address this problem, opportunistic routing (OR) [2] has been proposed as a new routing paradigm. OR utilizes the broadcast advantage of wireless communication to increase the reliability of a single transmission. Instead of relying on one next-hop node to forward a data packet (or simply, packet), OR pre-determines a set of candidate relays (or simply, relays) with a priority order and selects the highest-priority relay that indeed receives the packet as the actual forwarder based on the instantaneous channel conditions and node availability.

Although numerous variants of OR protocols [3]–[20] have been proposed for a variety of multi-hop wireless networks, none of the existing works have systematically studied the compatibility between routing metrics and routing protocols under the OR paradigm. Unscrupulously applying an arbitrary routing metric to an OR protocol may degrade network performance, such as routing loops, sub-optimality, or inconsistency.

Consider the ExOR model [2], where the expected transmission counts (ETXs) are used as a routing metric. ETX is actually the minimum expected number of transmissions required to achieve 100% transmission reliability for an unreliable link/path under the assumption of unlimited retransmissions. In ExOR, relays are prioritized in ascending order of their

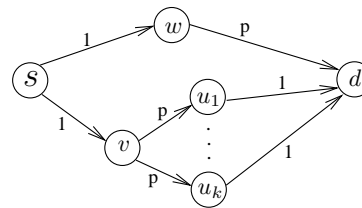


Fig. 1. An example network for ExOR. The number on each link represents the success probability of a single transmission.  $0 < p < 0.3$  and  $k > 1$ .

ETXs. The smaller the ETX, the higher the priority. While simulations [2] show EXOR's good performance, we can use the simple network topology shown in Fig. 1 to illustrate the problem of combining ETX with a link-state routing protocol using a Dijkstra-based algorithm. In this figure, each link is associated with a reliability value (success probability of a single transmission), which is equal to the reciprocal of its ETX value. When  $s$  uses the Dijkstra-based algorithm to calculate its OR scheme to  $d$ , it will incorrectly select a single path  $\langle s, w, d \rangle$  as the OR scheme with minimum ETX value. However, the actual minimum-ETX OR scheme is that in which  $s$  selects  $v$  as its only relay and  $v$  selects  $u_1, \dots, u_k$  as its relays because the ETX from  $v$  to  $d$  is greatly reduced due to  $v$ 's multiple relays. Hence, the incompatible combination of ETX and Dijkstra's algorithm creates a suboptimal routing decision. The details of computing the ETX-based OR scheme can be found in Section V-A.

The focus of this research is to provide a systematic analysis of the relationship between routing metrics and routing protocols in the OR paradigm. Our work identifies the basic properties that a routing metric needs to satisfy in order to work with different proactive or reactive OR protocols. The result of our research provides an important guideline for selecting the appropriate combinations of routing metrics and routing protocols under different communication environments. Based on our study, we consider several specific ORs as case studies.

The major contributions of this paper can be summarized as follows: 1) We design an OR algebra by exploring the nature of OR. 2) We identify several essential properties of our OR algebra. 3) We establish the relationship between the properties of our OR algebra and the requirements of OR protocols. 4) We use several existing routing metrics as case studies to show how to use our results to design OR protocols. 5) Our research results provide guidelines for designing OR protocols.

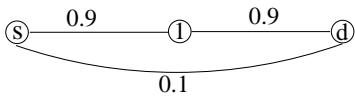


Fig. 2. An example of opportunistic routing. The number on each link represents the reliability of the link, i.e., the success probability of a single transmission over the link.

## II. PRELIMINARIES AND RELATED WORKS

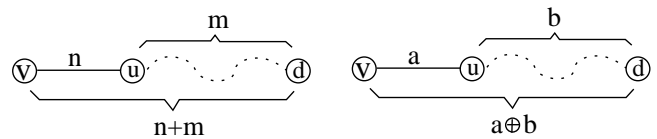
### A. Opportunistic Routing

The underlying idea behind OR is that a stable route is likely available in unreliable multi-hop wireless networks if each node can utilize its multiple neighbors as potential relays. For example, in Figure 2, although the link between  $s$  and  $d$  is highly unreliable, node 1 is likely to receive the packet sent by  $s$  and forward it to  $d$ . To find a stable and least-cost (in terms of ETX) route, ExOR [2] specifies two rules: the rule of the selection of a *relay set* for each node, and the rule of *relay prioritization*. Besides ExOR, numerous other OR protocols [3]–[20] have recently been proposed. For example, Dubois-Ferriere et al. [8] considered a least-cost OR scheme for wireless sensor networks with stringent energy constraints. Shah et al. [13] analyzed an OR model for the application environment of low traffic scenarios. Füllner et al. [9] studied an OR protocol for mobile wireless networks. Zorzi and Rao [20] applied the idea of OR to geographic routing. Chachulski et al. [5] integrated network coding into OR. Wu et al. [15] adopted OR in utility-based routing. However, none of the existing works provided a systematic analysis on the compatibility between routing objectives and diversified application environments in the OR framework.

### B. Routing Algebras

Routing algebras are best understood as a generalization of shortest-path routing as developed by Sobrinho [1]. Fig. 3 illustrates the similarity between shortest-path routing and routing algebra. In both shortest-path routing and routing algebra, paths are extended by incorporating unexplored links to known paths. The difference is that routing algebra generalizes link/path length in shortest-path routing to link label and path signature. The routing algebra [1] is a 5-tuple  $(\Sigma, \preceq, L, \oplus, \phi)$ , where  $\Sigma$  is a set of signatures for describing paths,  $\preceq$  is a preference relation over signatures,  $L$  is a set of labels for links,  $\oplus$  is a label application function mapping  $L \times \Sigma$  to  $\Sigma$ , and  $\phi$  is a special signature for a single node.

A preference relation conforms to completeness and transitivity, where completeness means that  $x \preceq y$  or  $y \preceq x$  (or both) for each  $x, y \in \Sigma$ , and transitivity denotes that  $x \preceq y$  and  $y \preceq z$  implies  $x \preceq z$  for each  $x, y, z \in \Sigma$ . The joint function and the preference relation have different implications for different metrics. Table I shows four concrete examples. Among them, the shortest-widest metric represents a path with the max-capacity among the set of paths with the minimum cost. The weight of this metric is in the form of  $(s, w)$ , where  $s$  represents the min-cost metric and  $w$  represents the max-capacity metric.  $(s_1, w_1) \oplus (s_2, w_2) = (s_1 + s_2, \min(w_1, w_2))$ .



(a) Path length computation. (b) Path signatures computation.

Fig. 3. Comparison of path computation for (a) shortest-path routing and (b) routing algebra, where dotted curves represent paths from  $u$  to  $d$ , and solid lines denote the link between  $u$  and  $v$ .

| metric    | C      | R        | W      | shortest-widest  |
|-----------|--------|----------|--------|--|
| $\oplus$  | +      | $\times$ | min    | $(s_1 + s_2, \min(w_1, w_2))$                            |
| $\preceq$ | $\leq$ | $\geq$   | $\geq$ | $(s_1 < s_2)$ or $(s_1 = s_2 \text{ and } w_1 \geq w_2)$ |

TABLE I

EXAMPLES OF THE ROUTING ALGEBRA. ‘C’ FOR MIN-COST (SHORTEST), ‘R’ FOR MAX-RELIABILITY, AND ‘W’ FOR MAX-CAPACITY (WIDEST).

In the preference relation, the min-cost metric has higher priority.  $(s_1, w_1) \preceq (s_2, w_2)$  if  $s_1 < s_2$  or  $s_1 = s_2$  and  $w_1 \geq w_2$ .

Sobrinho [1] designed a routing algebra to study the Dijkstra-based shortest path algorithm for Internet-based routing and inferred the relation between routing requirements (optimality and loop-freeness) and properties of the routing algebra. Later, Sobrinho [21] studied the routing algebra for the distributed Bellman-Ford algorithm and included routing convergence as another routing requirement. Yang and Wang [22] refined the properties of the routing algebra [1] and showed how these refined properties affect a wireless routing protocol’s ability to meet the three routing requirements: optimality, consistency, and loop-freeness. In this work, we design an OR algebra (based on the properties of OR), which is significantly different from existing routing algebra [1], [21], [22], and use it to analyze how the properties of a routing metric affect the routing requirements of an OR protocol.

## III. THE ALGEBRA FOR OPPORTUNISTIC ROUTING

### A. Network Model

We consider a set  $V$  of  $n$  wireless devices (called nodes hereafter) deployed in a given area. Each node has a unique ID  $v_i \in \{v_1, v_2, \dots, v_n\}$ , where  $n = |V|$ . Each node has an omnidirectional antenna, and a single transmission of a node can be received by any node within its transmission range. The network is then modeled as a graph  $(V, E)$ , where  $V = \{v_1, v_2, \dots, v_n\}$  is the set of nodes and  $E$  is the set of directional wireless links. A link  $(u, v)$  is in  $E$  if node  $v$  is in the transmission range of node  $u$ .

### B. Opportunistic Routing Algebra

Compared with traditional single-path routing and multi-path routing, OR can be regarded as any-path routing [19]. For any source-destination pair, OR pre-determines a set of candidate paths, but only one candidate path will be selected as the actual forwarding path. The path selection is dynamic and dependent on the instantaneous channel conditions and node availability. The set of candidate paths for a source-destination

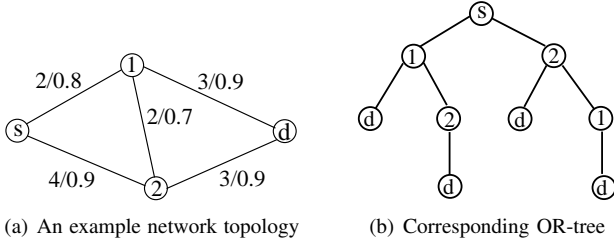


Fig. 4. An example network and its OR-tree for source  $s$  and destination  $d$ , where each link is associated two metrics: link cost/reliability.

pair can be regarded as a tree called *OR-tree*. For example, an OR-tree from  $s$  to  $d$  in Fig. 4(a) is shown in Fig. 4(b). We use  $S_{r(s)}(s, d)$  to represent the OR-tree from source  $s$  to destination  $d$  with  $r(s)$  representing  $s$ ' relay sequence (relays in a specific order). For example, in Fig. 4(b),  $r(s) = 12$  (node 1 preceding node 2). In the context where  $d$  and  $r(s)$  are explicit, we use  $S(s)$  for short.

In order to model opportunistic routing, we design an *opportunistic routing algebra*  $\mathcal{O}$  as follows:

$$\mathcal{O} = (\Sigma, \oplus, l, \preceq)$$

where  $\Sigma$  is the set of OR-trees,  $\oplus$  is a function that joints two OR-trees into a new OR-tree,  $l$  is a function that assigns a weight to an OR-tree, and  $\preceq$  is a preference relation over OR-trees in terms of weights.  $\forall S(v), S(u) \in \Sigma$ , it says  $S(v)$  is strictly preferred to  $S(u)$  (or  $S(v)$  has higher preference than  $S(u)$ ) if  $l[S(v)] \prec l[S(u)]$ .

$S(v) \oplus S(u)$  is interpreted as OR-tree  $S(v)$  merging OR-tree  $S(u)$  as its least-preference sub-OR-tree. It is illustrated by Fig. 5, where  $S(v)$  has already had two sub-OR-trees,  $S(a)$  and  $S(b)$ , before jointing  $S(u)$ . Note that existing sub-OR-trees are preferred to a newly jointed sub-OR-tree. In  $S(v) \oplus S(u)$ , the preference among three sub-OR-trees is  $l[S(a)] \preceq l[S(b)] \preceq l[S(u)]$ . Sub-OR-trees are ordered from left to right in descending order of preference.

Unlike the joint operation in [1], which simply merges two paths into a single path, our joint operation does not satisfy the commutation and association laws, i.e.,  $S(v) \oplus S(u) \neq S(u) \oplus S(v)$  and  $(S(v) \oplus S(u)) \oplus S(w) \neq S(v) \oplus (S(u) \oplus S(w))$ , as shown in Fig. 6. Three concrete examples of our joint operation can be found in Section V.

### C. Properties of the OR Algebra

Similar to the shortest-path routing, whether an OR protocol can satisfy various routing requirements depends on different properties of the OR algebra. In this section, we abstract and summarize the properties of our OR algebra.

The first property specifies the criterion for a node to select its relays. Intuitively, a node  $v$  selects a node  $u$  as its relay only if  $u$  can enhance  $v$ 's performance to forward packets to the destination. Based on the notations of our OR algebra, it can be rephrased as  $S(v)$  jointing  $S(u)$  only if  $l[S(v) \oplus S(u)] \prec l[S(v)]$ . In this case,  $u$  is said to be *beneficial* to  $v$ . Note that relays can only be selected from neighbors. We use  $N(v)$

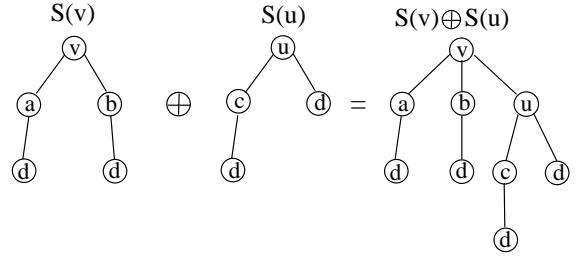


Fig. 5. The illustration of the  $\oplus$  operation in the OR algebra.

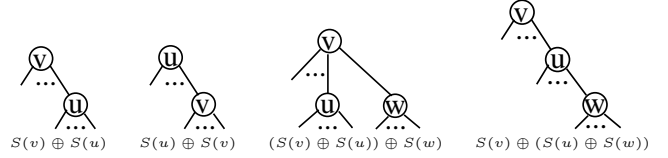


Fig. 6. Illustrate that OR algebra does not satisfy commutation law and association law.

to denote  $v$ 's neighbors and  $R(v)$  to denote  $v$ 's relay set. In Definitions 1 and 2, we assume that  $S(v)$  is the OR-tree before  $u$  is added into  $v$ 's relay set.

**Definition 1: (Relay beneficial condition)** In the OR algebra  $\mathcal{O}$ , relay  $u$  is conditionally beneficial to node  $v$  if  $l[S(v) \oplus S(u)] \prec l[S(v)]$  implies  $l[S(u)] \prec l[S(v)]$  and vice versa.  $\mathcal{O}$  is relay conditionally beneficial if  $\forall v \in V$  and  $\forall u \in R(v)$ , node  $u$  is conditionally beneficial to  $v$ .

Intuitively, if an OR protocol satisfies the relay-beneficial-condition, any node under this protocol can only select neighbors with higher preference as its relays. Note that not all OR are relay conditionally beneficial as illustrated in the utility-based OR [15]. If Definition 1 does not satisfy, the OR-tree in Fig. 4 (b) is possible. The second property compares the weight of an OR-tree with its newly jointed sub-OR-tree. If the sub-OR-tree is preferred to the OR-tree before the joint operation, the sub-OR-tree should be still preferred to the OR-tree that has jointed the sub-OR-tree.

**Definition 2: (Preference preservation)** The OR algebra  $\mathcal{O}$  is preference-preservable if  $l[S(u)] \preceq l[S(v)]$  implies  $l[S(u)] \preceq l[S(v) \oplus S(u)]$  for  $\forall v \in V, \forall u \in N(v)$ . Similarly,  $\mathcal{O}$  is strictly preference-preservable if  $l[S(u)] \prec l[S(v)]$  implies  $l[S(u)] \prec l[S(v) \oplus S(u)]$ .

The preference-preservation property can also be rephrased as  $l[S(v) \oplus S(u)] \prec l[S(u)]$  implying  $l[S(v)] \prec l[S(u)]$ . The third and fourth properties preserve the preference among OR-trees in a different manner. The third (fourth) property states that  $S(v)$  preferred to  $S(u)$  implies  $S(v)$  still preferred to  $S(u)$  if both of them joint (are jointed by) a common third OR-tree. The third (fourth) property resembles the right (left)-isotonicity introduced in [22], which states that the preference of two paths is preserved if both of them are appended (prefixed) by a common third path. Thus, we name the third (fourth) property quasi-right (left)-isotonicity.

**Definition 3: (Quasi-right-isotonicity)** The OR algebra  $\mathcal{O}$  is quasi-right-isotonic if  $l[S(v)] \preceq l[S(u)]$  implies  $l[S(v) \oplus S(w)] \preceq l[S(u) \oplus S(w)]$  for  $\forall S(w), S(v), S(u) \in \Sigma$  satisfying  $w \in N(v) \cap N(u)$ . Similarly,  $\mathcal{O}$  is strictly quasi-right-isotonic if  $l[S(v)] \prec l[S(u)]$  implies  $l[S(v) \oplus S(w)] \prec l[S(u) \oplus S(w)]$ .

**Definition 4: (Quasi-left-isotonicity)** The OR algebra  $\mathcal{O}$  is quasi-left-isotonic if  $l[S(v)] \preceq l[S(u)]$  implies  $l[S(w) \oplus S(v)] \preceq l[S(w) \oplus S(u)]$  for  $\forall S(w), S(v), S(u) \in \Sigma$  satisfying  $v, u \in N(w)$ . Similarly,  $\mathcal{O}$  is strictly quasi-left-isotonic if  $l[S(v)] \prec l[S(u)]$  implies  $l[S(w) \oplus S(v)] \prec l[S(w) \oplus S(u)]$ .

In Definitions 3 and 4,  $v$  and  $u$  can be the same node but their corresponding OR-trees are different. If the joint operation is idempotent, Definitions 1 and 2 can be regarded as special cases of Definitions 3 and 4. However,  $\oplus$  is not necessarily idempotent as illustrated by the utility-based OR model [15]. The fifth property specifies the criterion for a node to prioritize its relays. The priority of relays is based on the preference of associated OR-trees. The higher the preference, the higher the priority. We use  $S_{R(v)}(v)$  to represent the set of OR-trees rooted at  $v$  with  $R(v)$  being  $v$ 's relay set. Once all the sub-OR-trees of an OR-tree are determined, the OR-tree is uniquely determined by the sequence of relays associated with the sub-OR-trees. Hence, different relay sequences  $r(v)$  correspond to different OR-trees  $S_{r(v)}(v)$ . We also use  $\bar{r}(v)$  to represent the sequence of relays in  $R(v)$  in descending order of preference. For example, in Fig. 4(a),  $\bar{r}(s) = 12$ .  $S_{\bar{r}(s)}(s)$  is uniquely determined, as shown in Fig. 4 (b), if sub-OR-trees  $S(1)$  and  $S(2)$  are determined.  $S_{r(s)}(s)$  with  $r(s) = 21$  is a totally different OR-tree.

**Definition 5: (Relay-order-optimality)** A node is relay-order-optimal if the sufficient and necessary condition to optimize the weight of its associated OR-tree with respect to a given relay set is to order its relays in descending order of the preferences of associated sub-OR-trees. The OR algebra  $\mathcal{O}$  is relay-order-optimal if every node is relay-order-optimal, i.e.,  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{r(v)}(v)], \forall v \in V$  and  $\forall S_{r(v)}(v) \in S_{R(v)}(v)$ .

Since the relay-order-optimality property is relatively hard to verify for a given metric, we break it into two sub-properties and prove their equivalence by Lemma 1. One sub-property is a special case of the relay-order-optimality, which considers the order-optimality for only two relays.

**Definition 6: (2-relay-order-optimality)** A node  $v$  is 2-relay-order-optimal if  $l[S(u)] \preceq l[S(w)]$  implies  $l[S(v) \oplus S(u) \oplus S(w)] \preceq l[S(v) \oplus S(w) \oplus S(u)]$  for  $\forall u, w \in N(v)$  and vice versa. The OR algebra  $\mathcal{O}$  is 2-relay-order-optimal if every node is 2-relay-order-optimal.

Note that  $S_{r(v)uw}(v) = S_{r(v)} \oplus S(u) \oplus S(w)$ . The other sub-property is quasi-right-isotonicity. Without loss of generality, we can assume  $R(v) = \{u_1, \dots, u_k\}$  with  $l[S(u_i)] \preceq l[S(u_{i+1})]$  for  $i = 1, 2, \dots, k-1$ . We also define  $R_i(v) = \{u_1, u_2, \dots, u_i\}$  ( $1 \leq i \leq k$ ) ( $R_i^{-m}(v) = R_i(v) \setminus \{u_m\}$  ( $1 \leq m \leq i$ )),  $r_i(v)$  ( $r_i^{-m}(v)$ ) as a permutation of  $R_i(v)$  ( $R_i^{-m}(v)$ ), and  $\bar{r}_i(v)$  ( $\bar{r}_i^{-m}(v)$ ) as the permutation of  $R_i(v)$

( $R_i^{-m}(v)$ ) in descending order of preference. We have the following lemma to prove the claimed equivalence.

**Lemma 1:** The OR algebra is relay-order-optimal iff it is quasi-right-isotonic and 2-relay-order-optimal.

*Proof:* Sufficiency is proved by induction on the relay set size  $k$ . The inductive basis ( $k = 1$  and  $k = 2$ ) is trivial (directly from Definitions 5 and 6). For the inductive step, assume that the OR algebra is relay-order-optimal for any relay set with size  $k = i$ . Consider any relay set  $R(v)$  with  $k = i + 1$  for any node  $v$ . Since any permutation  $r(v)$  of  $R(v)$  can be regarded as the sequence of the first  $i$  relays appended by the last relay,  $r(v)$  can be represented by  $r^{-m}(v)u_m$  if we use  $u_m$  to denote the last relay. Because  $u_m$  can be any relay in  $R(v)$ , proving  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{r^{-m}(v)u_m}(v)]$  ( $m = 1, \dots, i + 1$ ) is equal to proving the objective,  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{r(v)}(v)]$ . Note that  $S_{r^{-m}(v)u_m}(v)$  can be written as  $S_{r^{-m}(v)}(v) \oplus S(u_m)$ .

By the inductive hypothesis, we have  $l[S_{\bar{r}^{-m}(v)}(v)] \preceq l[S_{r^{-m}(v)}(v)]$ . According to the quasi-right-isotonicity, we have  $l[S_{\bar{r}^{-m}(v)}(v) \oplus S(u_m)] \preceq l[S_{r^{-m}(v)}(v) \oplus S(u_m)]$ . Hence, to prove the objective, it is sufficient to prove  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{r^{-m}(v)}(v) \oplus S(u_m)]$ . Note that  $S_{\bar{r}(v)}(v)$  and  $S_{\bar{r}^{-m}(v)}$  can be written as  $S_{\bar{r}_i(v)}(v) \oplus S(u_{i+1})$  and  $S_{\bar{r}_i^{-m}(v)}(v) \oplus S(u_{i+1})$ , respectively. If  $m = i + 1$ , it automatically holds. If  $m < i + 1$ , we have  $l[S_{\bar{r}_i(v)}(v)] \preceq l[S_{\bar{r}_i^{-m}(v)}(v) \oplus S(u_m)]$  due to inductive hypothesis. Thus,  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{\bar{r}_i^{-m}(v)}(v) \oplus S(u_m) \oplus S(u_{i+1})]$  because of quasi-right-isotonicity. By the 2-relay-order-optimality property, we also have  $l[S_{\bar{r}_i^{-m}(v)}(v) \oplus S(u_m) \oplus S(u_{i+1})] \preceq l[S_{\bar{r}_i^{-m}(v)}(v) \oplus S(u_{i+1}) \oplus S(u_m)]$ . Therefore,  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{r^{-m}(v)}(v) \oplus S(u_m)]$  holds and the sufficiency is proved.

The necessity of the 2-relay-order-optimality is obvious since it is a special case of the relay-order-optimality. The necessity of the quasi-right-isotonicity is proved by contradiction. If the OR algebra is not quasi-right-isotonic,  $v$  and  $r_i(v)$  exist such that  $l[S_{\bar{r}_i(v)}(v)] \preceq l[S_{r_i(v)}(v)], \exists S_{r_i(v)}(v) \in S_{R_i(v)}(v)$ , but  $l[S_{r_i(v)}(v) \oplus S(u_{i+1})] \prec l[S_{\bar{r}_i(v)}(v) \oplus S(u_{i+1})]$ . This contradicts the definition of relay-order-optimality. ■

A specific example that satisfies the relay-order-optimality property but not the relay-beneficial-condition and the preference-preservation property will be shown in Section V-B. The last property specifies a phenomenon where all relays of a node are strictly preferred to the node.

**Definition 7: (Monotonicity)** The OR algebra  $\mathcal{O}$  is monotonic if all relays of any node are preferred to that node, i.e.,  $l[S(u)] \preceq l[S_{r(v)}(v)], \forall v \in V, \forall u \in R(v)$ , and  $\forall S_{r(v)}(v) \in S_{R(v)}(v)$ . Similarly, OR algebra  $\mathcal{O}$  is strictly monotonic if  $l[S(u)] \prec l[S_{r(v)}(v)]$ .

**Lemma 2:** The OR algebra is (strictly) monotonic if it is relay conditionally beneficial, (strictly) preference-preservable and relay-order-optimal.

*Proof:* Since the OR algebra is relay-order-optimal, we have  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{r(v)}(v)], \forall v \in V$  and  $\forall S_{r(v)}(v) \in S_{R(v)}(v)$ . Because a node will only select a beneficial relay, we

have  $l[S_{\bar{r}(v)}(v)] \prec l[S_{\bar{r}_{k-1}(v)}(v)]$ . Due to the relay-beneficial-condition, we have  $l[S(u_k)] \prec l[S_{\bar{r}_{k-1}(v)}(v)]$ . According to the (strict) preference-preservation property, we have  $l[S(u_k)] \preceq l[S_{\bar{r}(v)}(v)]$  ( $l[S(u_k)] \prec l[S_{\bar{r}(v)}(v)]$ ). Since  $l[S(u_i)] \preceq l[S(u_k)]$  ( $i = 1, \dots, k$ ), the sufficiency can be easily inferred. ■

Among the three sufficient conditions, the relay-beneficial-condition and the (strictly) preference-preservation property are necessary, but the relay-order-optimality property is not. We illustrate their necessity as follows. First, assume that only the relay-beneficial-condition is not satisfied. Thus, a  $u_{i+1}$  exists such that  $l[S_{\bar{r}_{i+1}(v)}(v)] \prec l[S_{\bar{r}_i(v)}(v)]$  but  $l[S_{\bar{r}_i(v)}(v)] \preceq l[S(u_{i+1})]$ . Hence,  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{\bar{r}_{i+1}(v)}(v)] \prec l[S(u_{i+1})]$  contradicts the definition of monotonicity. Second, assume that only the property of (strictly) preference-preservation does not hold. Thus, a  $u_{i+1}$  exists such that  $l[S(u_{i+1})] \preceq l[S_{\bar{r}_i(v)}(v)]$  ( $l[S(u_{i+1})] \prec l[S_{\bar{r}_i(v)}(v)]$ ) but  $l[S_{\bar{r}_{i+1}(v)}(v)] \prec l[S(u_{i+1})]$  ( $l[S_{\bar{r}_{i+1}(v)}(v)] \preceq l[S(u_{i+1})]$ ). Since  $l[S_{\bar{r}(v)}(v)] \preceq l[S_{\bar{r}_{i+1}(v)}(v)]$ , it also contradicts the definition of (strictly) monotonicity. Third, the relay-order-optimality is not necessary, since the OR algebra can be monotonic, even if it is not relay-order-optimal.

In the end, we specify the necessary and sufficient condition to ensure that every sub-OR-tree of an optimal OR-tree is optimal. This is specified by Lemma 3, where we use  $S^*(s)$  to denote the optimal OR-tree from  $s$  to  $d$ .

*Lemma 3:* Every sub-OR-tree of an optimal OR-tree is optimal iff the OR algebra is strictly quasi-left-isotonic.

*Proof:* Sufficiency proved by contradiction. Assume that an optimal OR-tree  $S^*(s)$  exists such that at least one of its sub-OR-trees is not optimal. Assume that  $S(v)$  is the first non-optimal sub-OR-tree. Note that order is first from top to bottom and then from left to right. Let  $u$  be a parent node of  $v$  on  $S^*(s)$  and  $S(u)$  denote  $u$ 's associated OR-tree before jointing  $S(v)$ . Due to the strict quasi-left-isotonicity property, we have  $l[S(u) \oplus S^*(v)] \prec l[S(u) \oplus S(v)]$  since  $l[S^*(v)] \prec l[S(v)]$ . Hence, the sub-OR-tree associated with  $u$  is not optimal, contradicting our assumption. The sufficiency is thus proved.

Necessity also proved by contradiction. Assume that the OR algebra is not strictly quasi-left-isotonic. A possibility exists that  $l[S(s) \oplus S(v)] \preceq l[S(s) \oplus S'(v)]$  but  $S'(v) \prec S(v)$ . Hence, the sub-OR-tree may not be the optimal OR-tree - a contradiction. The necessity is thus proved. ■

#### IV. APPLICATION OF THE OR ALGEBRA

Numerous routing requirements are necessary to ensure efficient and robust routing operations. These requirements include (but are not limited to) optimality, loop-freeness, and consistency. In this section, we first define these three requirements. We then discuss the properties required for the OR algebra to meet these requirements for two different OR algorithms.

---

#### Algorithm 1 Dijkstra-Based Opportunistic Routing Algorithm

---

```

Dijkstra( $G, d$ )
1:  $Q \leftarrow V$ ;
2: while  $Q \neq \emptyset$  do
3:    $u \leftarrow \text{Extract-Best}(Q)$  {node to visit};
4:   for each  $v \in N(u) \cap Q$  do
5:     if  $l[S(v) \oplus S(u)] \prec l[S(v)]$  then
6:        $S(v) \leftarrow S(v) \oplus S(u)$ ;

```

---

#### A. Requirements for Opportunistic Routing Protocols

*Definition 8: (Optimality)* An OR protocol is optimal if it always routes packets along the highest-preference OR-tree between every pair of nodes in any connected network.

*Definition 9: (Consistency)* An OR protocol is consistent if, for any source and destination pair  $(s, d)$  and any node  $v$  on the OR-tree  $S(s, d)$ ,  $v$  forwards packets from  $s$  to  $v$  and from  $v$  to  $d$  always along  $S(s, d)$ .

*Definition 10: (Loop-freeness)* An OR protocol is loop-free if it does not create any packet forwarding loop for any connected network.

#### B. Opportunistic Routing Using Dijkstra's Algorithm

Numerous opportunistic routing protocols depend on Dijkstra-based algorithms to compute the opportunistic routing schemes from all the other nodes to the destination. We presented the general Dijkstra-based OR algorithm in Algorithm 1, where procedure Extract-Best extracts the highest-preference node from  $Q$ . Each time a node has been extracted, it will relax its neighbors remaining in  $Q$  by providing itself as a relay candidate. A relay will be inserted into the relay set of a node only if the relay is beneficial to the node's OR-tree.

As mentioned in the introduction, the optimality of the OR scheme depends on the combination of the underlying OR algorithm and the associated routing metric. Theorem 1 presents the sufficient properties that a routing metric should have to guarantee optimality when combined with the Dijkstra-based OR algorithm. The proof of Theorem 1 needs the following lemma, where  $v_i$  denotes the  $i$ -th node extracted from  $Q$  and  $S(v_i)$  represents the associated OR-tree after  $v_i$  is extracted from  $Q$ . Lemma 4 specifies the sufficient and necessary conditions that the preferences of associated OR-trees decrease in the order of nodes being extracted from  $Q$ .

*Lemma 4:* In Algorithm 1,  $l[S(v_i)] \preceq l[S(v_{i+1})]$  ( $i = 1, \dots, |V| - 1$ ) iff the OR algebra is preference-preservable.

*Proof:* Sufficiency proved by contradiction. Assume that  $v_j$  is the first node satisfying  $l[S(v_{j+1})] \prec l[S(v_j)]$ . Right before  $v_j$  is extracted from  $Q$ , we have  $l[S(v_j)] \preceq l[S'(v_{j+1})]$ , where  $S'(v_{j+1})$  represents the OR-tree associated with  $v_{j+1}$  at that moment. Otherwise,  $v_{j+1}$  will be extracted before  $v_j$ . Between  $v_j$ 's and  $v_{j+1}$ 's extractions from  $Q$ ,  $v_j$  is the only

available relay candidate for  $v_{j+1}$ . If  $v_{j+1}$  does not select  $v_j$ ,  $S(v_{j+1}) = S'(v_{j+1})$ , and thus  $l[S(v_j)] \preceq l[S(v_{j+1})]$ , a contradiction. Otherwise, we have  $S(v_{j+1}) = S'(v_{j+1}) \oplus S(v_j)$ . By the assumption, we have  $l[S'(v_{j+1}) \oplus S(v_j)] \prec l[S(v_j)]$ . Because the OR algebra is preference-preservable, we have  $l[S'(v_{j+1})] \prec l[S(v_j)]$ , a contradiction. Hence, the sufficiency is proved.

Necessity also proved by contradiction. Assume that the OR algebra is not preference-preservable. Hence,  $\exists S(v), S(u) \in \Sigma$  s.t.  $l[S(u)] \preceq l[S(v)]$  and  $l[S(v) \oplus S(u)] \prec l[S(u)]$ . Therefore, it is possible for some  $v_j$  and  $v_{j+1}$  satisfying  $l[S(v_j)] \preceq l[S'(v_{j+1})]$  and  $l[S'(v_{j+1}) \oplus S(v_j)] \prec l[S(v_j)]$ , a contradiction. The necessity is thus proved. ■

In Algorithm 1, when a node  $u$  is going to be extracted from  $Q$ , its associated OR-tree has the highest preference among all OR-trees of remaining nodes in  $Q$ . However, it is possible that the associated OR-tree of a remaining node  $v$  in  $Q$  will have a higher preference later so that  $l[S(v)] \prec l[S(u)]$ . Lemma 4 specifies a necessary and sufficient condition that the above situation will not occur. In the proof of Theorem 1, we use  $R^*(v)$  to denote  $v$ 's optimal relay set associated with optimal OR-tree  $S^*(v)$ , and define  $\delta(v) = l[S^*(v)]$ .

*Theorem 1:* The Dijkstra-based OR algorithm can find the optimal OR scheme if the combined OR algebra is relay-conditionally-beneficial, strictly preference-preservable, and relay-order-optimal.

*Proof:* To prove optimality, it is sufficient to prove that  $l[S_{R(v)}(v)] = \delta(v)$  for any node  $v$  when  $v$  is extracted from  $Q$  because the OR-tree associated with a node will not change after its extraction. We prove via induction. The inductive basis is trivial since  $l[S(d)] = \delta(d)$  when  $d$  is extracted from  $Q$ . For the inductive step, assume  $l[S_{R(w)}(w)] = \delta(w)$  for all nodes  $w$  that have been extracted from  $Q$  before  $v$ . For contradiction, suppose  $\delta(v, d) \prec l[S_{R(v)}(v)]$  when  $v$  is extracted from  $Q$ .

From Algorithm 1 and Lemma 4, it can be inferred that  $v$ 's extracted neighbors are inserted into  $R(v)$  in descending order of their preferences until  $v$  itself is extracted from  $Q$ . Hence, when  $v$  is extracted from  $Q$ ,  $l[S_{R(v)}(v)]$  is optimized with respect to  $R(v)$  due to the relay-order-optimality property. According to the inductive hypothesis, every sub-OR-tree  $S(u_i)$  ( $\forall u_i \in R(v)$ ) should be optimal. Hence, the preference of  $S_{R(v)}(v)$  cannot improve by increasing the preference of  $S(u_i)$ . Based on Line 5 of Algorithm 1, every relay in  $R(v)$  is beneficial. Therefore, the preference of  $S_{R(v)}(v)$  cannot improve by removing relays from  $R(v)$ . Thus, we can conclude that the optimal relay set  $R^*(v)$  is not a strict subset of  $R(v)$ . Hence, at least one beneficial relay in  $R^*(v)/R(v)$  must exist if  $\delta(v) \prec l[S_{R(v)}(v)]$ .

Without loss of generality, assume that  $u$  is the relay in  $R^*(v)/R(v)$ , the OR-tree of which has the highest preference. Let  $S(u)$  denote the sub-OR-tree of  $S^*(v)$  and  $x$  be the first node along  $S(u)$  that is still in  $Q$  before  $v$  is extracted. Note that the existence of  $u$  relies on the existence of  $x$ , which can also be  $u$ . According to Lemma 2, the OR algebra is strictly

monotonic. Hence, for any  $u' \in S^*(v)$ , we have  $l[S(u')] \prec l[S^*(v)]$ , especially,  $l[S(x)] \prec l[S^*(v)]$ . Thus,  $x$  should be extracted from  $Q$  before  $v$ , which is a contradiction. Hence,  $R(v) = R^*(v)$  and  $l[S_{R(v)}(v)] = \delta(v)$  when  $v$  is extracted from  $Q$ . ■

If the OR computation is executed in a centralized way, such as source routing, loop-freeness and consistency are automatically guaranteed. For distributed OR computation, loop-freeness and consistency require the properties of the OR algebra as shown in Theorem 2 and Theorem 3, respectively. Note that the sufficient conditions for loop-freeness and optimality are in fact the same, while the sufficient conditions for consistency are stronger than those for optimality.

*Theorem 2:* The distributed OR based on Dijkstra-based OR algorithm is loop-free if the OR algebra is strictly monotonic.

*Proof:* We use contradiction to prove this theorem. Assume that a routing loop exists while packets are forwarded from source  $s$  to destination  $d$  in the OR-tree. Denote the first node appearing in the routing loop along OR-tree  $S(s)$  as  $v$ . To cause a routing loop, in the OR-tree  $S(v)$ , at least one node  $w_1$  (other than  $v$  and  $d$ ) must deviate from  $S(v)$ . In  $S(w_1)$ , another node  $w_2$  might also deviate from  $S(w_1)$ . This process continues until some node  $w_k$  deviating from  $S(w_{k-1})$  reaches  $v$  along  $S(w_k)$ . Since the OR algebra is strictly monotonic, according to Lemma 2, we have  $l[S(v)] \prec l[S(w_k)] \prec \dots \prec l[S(w_1)] \prec l[S(v)]$ , a contradiction. Thus, this theorem is proved. ■

*Theorem 3:* The distributed OR based on Dijkstra's algorithm is consistent if the OR algebra is relay-conditionally-beneficial, strictly preference-preservable, relay-order-optimal, strictly quasi-left-isotonic, and no two OR-trees have the same preference.

*Proof:* The relay-beneficial-condition, strict preference-preservation, and relay-order-optimality guarantee that the Dijkstra-based OR algorithm can always find the optimal OR-tree between source  $s$  and destination  $d$  (Theorem 1). We prove consistency by contradiction. According to the definition of consistency, an OR protocol is inconsistent only if at least one node on  $S(s)$  might not forward packets along  $S(s)$ . Assume  $v$  is such a node on  $S(s)$ . Since both  $s$  and  $v$  will forward packets along their optimal OR-trees, we have  $S(v) \neq S^*(v)$ , where  $S(v)$  is a sub-OR-tree of  $S^*(s)$ . Note that  $l[S^*(v)] \prec l[S(v)]$  because no two OR-trees have the same preference. Because of the strict quasi-left-isotonic and Lemma 3, we also have  $l[S(v)] \prec l[S^*(v)]$ , a contradiction. Therefore, no such node  $v$  exists. The theorem is thus proved. ■

### C. Opportunistic Routing Using Bellman-Ford Algorithm

Another algorithm to compute the OR scheme is the Bellman-Ford based algorithm, which can compute the OR scheme for all source-destination pairs. We present a Bellman-Ford algorithm specified for the OR scheme as shown in

---

**Algorithm 2** Bellman-Ford-Based OR Algorithm

---

**RelayCompute**( $v$ )

- 1: Sort  $u_j \in N(v)$  in descending order of preference;
- 2: Initialize:  $S(v) \leftarrow S_\emptyset(v)$ ;  $j \leftarrow 1$ ;
- 3: **while**  $l[S(v) \oplus S(u_j)] \prec l[S(v)]$  **do**
- 4:      $S(v) \leftarrow S(v) \oplus S(u_j)$ ;
- 5:      $j \leftarrow j + 1$ ;
- 6: **return**  $l[S(v)]$ ;

**MainProgram**

- 1: Initialize:  $l[S(v_i)] \leftarrow \infty$ ,  $l[S(d)] \leftarrow 0$ ;
  - 2: **repeat**
  - 3:     Sort nodes in descending order of preference;
  - 4:     **for**  $i = 2$  to  $|V|$  **do**
  - 5:          $l[S(v_i)] \leftarrow \text{RelayCompute}(v_i)$ ;
  - 6: **until** no further update
- 

Algorithm 2. To simplify the presentation, we consider a fixed destination  $d$  for this Bellman-Ford algorithm. It is easy to extend the algorithm to all source-destination pairs.

In line 3 of the main program of Algorithm 2, all nodes are sorted in the descending order of their weight preferences and labeled as  $v_i$  with smaller subscripts representing higher preference. The tie-breaking rule is that the node with smaller ID will have higher preference if two nodes are equally preferred.  $v_1$  represents destination  $d$ , which is not counted in the update process. Each time when a node calls procedure RelayCompute, it will empty its relay set and re-insert its relays in the descending order of their preferences.

*Theorem 4:* The Bellman-Ford-based OR algorithm can find the optimal OR scheme if the OR algebra is relay-conditionally-beneficial, strictly preference-preservable, and relay-order-optimal.

*Proof:* It is sufficient to prove that in the MainProgram of Algorithm 2 the first  $m$  nodes with higher preference in the nodes sorted by line 3 converge to their optimality after  $m - 1$  iterations of the outer loop. We prove this by induction. The inductive basis is trivial since the first node is the destination, which has the highest preference. For the inductive step, assume that the claim holds for the first  $m - 1$  iterations. During the  $m$ -th iteration, the first  $m$  nodes will not update their preferences and corresponding relays since they have converged to their optimality, while the remaining  $n - m$  nodes will insert beneficial neighbors in descending order of their preferences. After the  $m$ -th iteration, one node, denoted as  $v_{m+1}$ , will have the highest preference among the remaining  $n - m$  nodes. Because  $v_{m+1}$  ranks  $(m + 1)$ -th in terms of preference, it selects relays only from the first  $m$  nodes according to the strict monotonicity property. Hence,  $\forall u_i \in R(v_{m+1})$ ,  $S(u_i)$  is optimal due to the inductive hypothesis. Based on the relay-order-optimality property, it can be inferred that  $S_{\bar{r}(v_{m+1})}(v_{m+1})$  is optimal with respect to  $S^*(u_i)$  for all  $u_i \in R(v_{m+1})$ . Since every relay is beneficial, the preference of  $S_{\bar{r}(v_{m+1})}(v_{m+1})$  cannot improve by removing

relays from  $R(v_{m+1})$ . Thus, we can conclude that the optimal relay set  $R^*(v_{m+1})$  is not a strictly subset of  $R(v_{m+1})$ . Hence, at least one beneficial relay in  $R^*(v_{m+1})/R(v_{m+1})$  must exist if  $\delta(v_{m+1}) \prec l[S_{\bar{r}(v_{m+1})}(v_{m+1})]$ . Without loss of generality, assume that  $u$  is the relay in  $R^*(v_{m+1})/R(v_{m+1})$ , the OR-tree of which has the highest preference. Let  $S(u)$  denote the sub-OR-tree of  $S^*(v_{m+1})$  and  $w$  be the first node in  $S(u)$ , which is still in the  $n - m$  remaining nodes, satisfying  $l[S_{\bar{r}(v_{m+1})}(v_{m+1})] \preceq l[S(w)]$ . Note that  $w$  exists because  $u \notin R(v_{m+1})$ . Since the OR algebra is strict monotonic, for any  $u' \in S^*(v_{m+1})$ , we have  $l[S(u')] \prec l[S^*(v_{m+1})]$ , especially,  $l[S(w)] \prec l[S^*(v_{m+1})]$ , a contradiction (Note  $l[S^*(v_{m+1})] \preceq l[S_{\bar{r}(v_{m+1})}(v_{m+1})]$ ). Therefore,  $l[S_{R(v_{m+1})}(v_{m+1})] = \delta(v_{m+1})$  after  $m$  iterations. ■

## V. CASE AND EXPERIMENT STUDY

In previous sections, we analyzed the sufficient and necessary conditions to maintain loop-freeness, optimality, and consistency for Dijkstra's routing protocols. In this section, we use several existing routing metrics to demonstrate how to use our research results to design OR protocols.

## A. ETX-based OR

The first example is the OR model called ExOR, which was proposed by Biswas and Morris [2] to minimize the expected transmission counts (ETX) in multi-hop wireless networks. In ExOR, the weight of an OR-tree  $S(v)$  is recursively defined as follows:

$$l[S(v)] = \min_{u_i \in R(v)} \{l[S(v)], l[S(u_i)] + ETX(v, u_i)\},$$

where  $ETX(v, u_i) = \frac{1}{p_{v, u_i}}$  is the expected number of transmissions along link  $(v, u_i)$  to guarantee reliable transmission. Here,  $p_{v, u_i}$  is the success probability of a single transmission along link  $(v, u_i)$ . The joint operation is defined as follows:

$$l[S(v) \oplus S(u)] = \min\{l[S(v)], l[S(u)] + ETX(v, u)\}.$$

An OR-tree  $S(v)$  is preferred to an OR-tree  $S(u)$ , i.e.,  $l[S(v)] \preceq l[S(u)]$  if  $l[S(v)]$  is smaller than or equal to  $l[S(u)]$ , i.e.,  $l[S(v)] \leq l[S(u)]$ . The ExOR model is preference-preservable, since  $l[S(u)] \leq \min\{l[S(v)], l[S(u)] + ETX(v, u)\}$  if  $l[S(u)] \leq l[S(v)]$ . It is also relay-order-optimal since the weight of an OR-tree does not depend on the relative order of its sub OR-trees. However, it is not relay-conditionally-beneficial since  $l[S(u)] < l[S(v)]$  does not imply  $\min\{l[S(v)], l[S(u)] + ETX(v, u)\} < l[S(v)]$  if  $l[S(v)] \leq l[S(u)] + ETX(v, u)$ . Hence, we can conclude that the Dijkstra-based OR algorithm cannot achieve optimal OR scheme by using the ETX metric.

However, if ExOR adopts the EOTX metric [23], the Dijkstra-based OR algorithm can find the optimal OR scheme. Assuming  $\bar{r}(v) = u_1 u_2 \cdots u_k$ , the EOTX of an OR-tree  $S(v)$  can be recursively defined as follows:

$$l[S(v)] = \frac{\sum_{i=1}^k l[S(u_i)] \cdot p_{v, u_i} \cdot \prod_{j=1}^{i-1} (1 - p_{v, u_j})}{1 - \prod_{i=1}^k (1 - p_{v, u_i})}.$$

|      | d | w   | $u_i$ | v     | s     |
|------|---|-----|-------|-------|-------|
| ETX  | 0 | 1/p | 1     | 1+1/p | 1+1/p |
| EOTX | 0 | 1/p | 1     | 1+1/q | 2+1/q |

TABLE II

THE ETX AND EOTX VALUES OF NODES IN FIG. 1 AFTER THE EXECUTION OF THE DIJKSTRA-BASED OR ALGORITHM.  $q = 1 - (1 - p)^k$ .

The corresponding joint operation can be defined as follows:

$$l[S(v) \oplus S(u)] = \frac{l[S(v)] \cdot (1 - P(v)) + l[S(u)] \cdot p_{v,u} \cdot P(v)}{1 - (1 - p_{v,u}) \cdot P(v)},$$

where  $P(v) = \prod_{i=1}^k (1 - p_{v,u_i})$ .

The EOTX metric is strictly preference-preservable and relay-conditionally-beneficial as  $l[S(u)] < l[S(v) \oplus S(u)] < l[S(v)]$  when  $l[S(u)] < l[S(v)]$  and  $0 < p_{v,u}, P(v) < 1$ . Hence, the combination of the EOTX metric and the Dijkstra-based OR algorithm can compute the optimal OR scheme.

We use Fig. 1 to illustrate the gap between the ETX and EOTX metrics. After the execution of the Dijkstra-based OR algorithm, the ETX and EOTX values of each node in Fig. 1 are shown in Table II. When  $p < 0.3$  and  $k > 1$ , the  $EOTX(s) < ETX(s)$  and  $\lim_{p \rightarrow 0} \frac{ETX(s)}{EOTX(s)} = k$ . Hence, the performance of ETX metric can be arbitrarily worse than that of EOTX.

### B. Utility-based OR

Wu et al. [15] proposed a utility-based OR model to balance the trade-off between transmission cost and transmission reliability. In their work, the weight of an OR-tree  $S(v)$  is recursively defined as:

$$l[S(v)] = \sum_{i=1}^k l[S(u_i)] \cdot p_{v,u_i} \cdot \prod_{j=1}^{i-1} (1 - p_{v,u_j}) - c, \quad (1)$$

where  $u_i$  and  $p_{v,u_i}$  are the same as defined in subsection V-A, and  $c$  represents the transmission cost of  $v$ . The value of  $c$  depends on  $v$ 's transmission range. The weight  $l[S(v)]$  essentially represents the expected utility for delivering a packet over OR-tree  $S(v)$ . Especially,  $l[S(d)]$  reflects the packet importance. The joint operation for a given value of  $c$  is defined as follows:

$$l[S(v) \oplus S(u)] = l[S(v)] + l[S(u)] \cdot p_{v,u} \cdot \prod_{i=1}^k (1 - p_{v,u_i}).$$

An OR-tree  $S(v)$  is preferred to an OR-tree  $S(u)$ , i.e.,  $l[S(v)] \geq l[S(u)]$  if  $l[S(v)]$  is larger than  $l[S(u)]$ , i.e.,  $l[S(v)] \geq l[S(u)]$ .

This utility-based OR scheme is relay-order-optimal as proved by Theorem 1 in [15], but it is not relay-conditionally-beneficial and preference-preservable. For example, in Figure 4(a), if  $l[S(d)] = 20$ , then  $l[S_d(1)] = l[S_d(2)] = 15$  and  $l[S_{d1}(2)] = 16.05$  as computed by Algorithm 1, assuming that the node with lower ID will be extracted from  $Q$  first in the case of a tie. It is easy to verify that  $l[S_{d1}(2)] = l[S_d(2) \oplus S_d(1)] < l[S_d(2)]$  but  $l[S_d(2)] = l[S_d(1)]$ , contradicting the

relay-beneficial-condition. Also, since  $l[S_d(2)] = l[S_d(1)]$  but  $l[S_d(2) \oplus S_d(1)] < l[S_d(1)]$ , it is not preference-preservable. Therefore, the Dijkstra-based OR algorithm cannot compute the optimal OR scheme for this utility-based OR model.

### C. Simulation

Previously, we applied our OR algebra to three concrete OR schemes, i.e., ETX, EOTX, and utility-based OR. Among them, the EOTX-based OR and the utility-based OR can be regarded as two extreme cases. The EOTX-based OR allows unlimited retransmissions, while the utility-based OR forbids any retransmission. Their difference on the retransmission restriction produces different results on routing optimality when combining with Dijkstra-based OR algorithm. In this simulation study, we will explore the relationship between the EOTX-based OR and the utility-based OR and the underlying reason of their optimality gap by evaluating the effect of retransmission counts. All simulations are conducted in our customized simulator.

1) *Simulation Environment*: We set up the simulation in a  $900m \times 900m$  area. We assume nodes are homogeneous and can be deployed in this area arbitrarily. We fix the position of source  $s$  and destination  $d$  at locations  $(50m, 450m)$  and  $(850m, 450m)$ , respectively, and randomly deploy the intermediate nodes. For each set of parameters, we run each algorithm 100 times and use the average value of the results to evaluate the performance. We set the maximum transmission range to  $300m$ . Since the transmission range is fixed, the connectivity of the network is controlled solely by the number of nodes.

2) *Simulation Results*: We first compare the EOTX metric with the utility metric. To be consistent, we transform the transmission cost in the utility-based OR to the transmission count by setting  $c = 1$  in Formula (1). The simulation results are shown in Figs. 7 (a) and (b), where  $l[S(d)] = 1000$  and the number of nodes varies from 10 to 100 in increments of 10. From Fig. 7 (a), we can conclude that increasing the upper bound of retransmissions can increase the reliability of the utility-based OR. However, the reliability cannot be 100% unless the retransmissions become unlimited. The reliability for EOTX is not 100% when the number of nodes is less than 30 because the network may not be connected when the number of nodes is too sparse. The increment of reliability is at the expense of retransmissions as shown in Fig. 7 (b). Hence, a trade-off exists between the increment of reliability and the increment of transmission counts. Utility-based OR can help balance this trade-off.

However, the Dijkstra-based OR algorithm for utility-based OR is not optimal as shown by the simulation result in Fig. 7 (c), where we compute the approximation ratio of the Dijkstra-based OR algorithm to the optimal solution (with exponential time complexity) proposed in [15]. We compare the approximation ratio for three values of  $l[S(d)]$  with the tunable parameter, retransmission upper bound, from 0 to 10 in increments of 1. The result shows that the approximation ratio does not depend on the value of  $l[S(d)]$ . As the upper bound of retransmissions increases, the approximation ratio



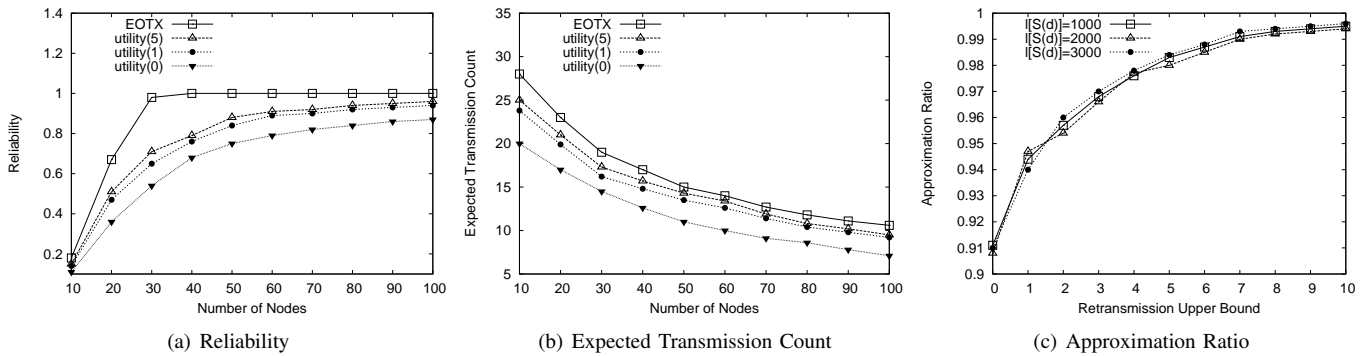


Fig. 7. Simulation results, where utility( $i$ ) in (a) and (b) represent that the upper bound of retransmissions is  $i$  in the utility-base OR.

will increase. However, the approximation ratio cannot be 1 unless the retransmissions become unlimited. The underlying reason is that the utility-based OR is not relay-conditionally-beneficial if the upper bound of retransmissions is a finite number. When the retransmission count does not reach its upper bound, a retransmission is strictly preferred to the nodes with lower preference. However, when it reaches its upper bound, nodes with a lower preference can provide additional transmission opportunities. Hence, the utility-based OR is not relay-conditionally-beneficial.

## VI. CONCLUSIONS

In this work, we have presented a systematic study of the design of a routing algebra for opportunistic routing. To the best of our knowledge, our research is the first to model opportunistic routing by means of a routing algebra, and identifies the relationship between the properties of a routing metric and the routing requirement of an OR protocol. Our work provides a systematic method of designing a proper routing metric for opportunistic routing. In the future, we will use our OR algebra to explore more OR algorithms and design new OR algorithms with fewer restrictions on the properties of routing metrics.

## ACKNOWLEDGEMENT

This work was supported in part by NSF grants CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

## REFERENCES

- [1] J. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the internet," in *Proceedings of IEEE INFOCOM 2001*, 2001, pp. 727–735.
- [2] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in *Proceedings of ACM SIGCOMM'05*, 2005, pp. 133–144.
- [3] J. Ai, A. A. Abouzeid, and Z. Ye, "Cross-layer optimal decision policies for spatial diversity forwarding in wireless ad hoc networks," in *Proceedings of the third IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS'06)*, 2006.
- [4] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection," *IEEE journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 659–672, 2006.
- [5] S. Chachulski, M. Jennings, S. K., and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proceedings of ACM SIGCOMM'07*, 2007.
- [6] R. R. Choudhury and N. H. Vaidya, "MAC-layer anycasting in ad hoc networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, 2004.
- [7] T. Cui, L. Chen, T. Ho, and S. Low, "Opportunistic source coding for data gathering in wireless sensor networks," in *Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'07)*, 2007.
- [8] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Least-Cost Opportunistic Routing," in *Proceedings of 2007 Allerton Conference on Communication, Control, and Computing*, 2007.
- [9] H. Fubler, J. Widmer, M. Kasemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Elsevier's Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.
- [10] M. Kurth, A. Zubow, and J. Redlich, "Cooperative opportunistic routing using transmit diversity in wireless mesh networks," in *Proceedings of IEEE INFOCOM 2008*, 2008.
- [11] P. Larsson, "Selection diversity forwarding in a multihop packet radio network with fading channel and capture," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, 2001.
- [12] T. Parker and K. Langendoen, "Guesswork: Robust routing in an uncertain world," in *Proceedings of the second IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS'05)*, 2005.
- [13] R. C. Shah, S. Wietholter, A. Wolisz, and J. M. Rabaey, "Modeling and analysis of opportunistic routing in low traffic scenarios," in *Proceedings of the Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WIOPT'05)*, 2005.
- [14] M. R. Souryal and N. Moayeri, "Channel-adaptive relaying in mobile ad hoc networks with fading," in *Proceedings of IEEE SECON 2005*, 2005.
- [15] J. Wu, M. Lu, and F. Li, "Utility-based opportunistic routing in multi-hop wireless networks," in *Proceedings of IEEE ICDCS 2008*, 2008.
- [16] Y. Yuan, H. Yang, S. Wong, S. Lu, and W. Arbaugh, "ROMER: Resilient opportunistic mesh routing for wireless mesh networks," in *Proceedings of IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005, 2005.
- [17] K. Zeng, W. Lou, and H. Zhai, "On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks," in *Proceedings of IEEE INFOCOM 2008*, 2008.
- [18] B. Zhao and M. C. Valenti, "Practical relay networks: A generalization of hybrid-ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, 2005.
- [19] Z. Zhong and S. Nelakuditi, "On the efficacy of opportunistic routing," in *Proceedings of the IEEE SECON 2007*, 2007.
- [20] M. Zorzi and R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," *IEEE Transaction on Mobile Computing*, vol. 2, no. 4, 2003. [Online]. Available: citeseer.ist.psu.edu/zorzi03energy.html
- [21] J. Sobrinho, "An algebraic theory of dynamic network routing," *IEEE/ACM Transactions on Networking*, pp. 1160–1173, 2005.
- [22] Y. Yang and J. Wang, "Design guidelines for routing metrics in multihop wireless networks," in *Proceedings of IEEE INFOCOM 2008*, 2008.
- [23] S. Chachulski, "Trading structure for randomness in wireless opportunistic routing," Master's thesis, MIT, 2007.