QoS Routing Using Dominant-Distance Vectors

J.J. Garcia-Luna-Aceves, Bradley R. Smith, Judith T. Samson
Computer Science and Engineering Department
University of California, Santa Cruz
Santa Cruz, CA, USA
{jj, brad, jtsamson}@soe.ucsc.edu

***Abstract*—The Dominant-Distance Routing Information Protocol (DRIP) is introduced for quality-of-service (QoS) routing based on multiple criteria and is proven to be loop-free at every instant and capable of converging to optimal routes if they exist. DRIP is based on the exchange of updates and queries stating reference routing-metric values for destinations. Simulation experiments based on ns3 are used to compare DRIP against the Non-Restarting Vectoring Protocol recently proposed by Sobrinho and Ferreira, as well as OSPF and RIPv2. The results demonstrate that DRIP provides loop-free routing based on multiple performance and policy criteria as efficiently as routing protocols for shortest-path routing.**

I. Introduction

The routing protocols used in the Internet and computer networks today and the vast majority of routing protocols developed since the advent of packet switching [1] assume a single criterion to determine the optimality of paths selected to reach destinations, which allows all path weights to be comparable to one another. Unfortunately, the performance characteristics of some paths may be inherently incomparable with one another in the Internet and in computer networks with heterogeneous links and multiple policy criteria for resource utilization. For example, a path with 100 Mbps bandwidth, delay of 0.5 msec, and 0.90 reliability is better than a path with 100 Mbps bandwidth, delay of 1 msec, and 0.90 reliability; however, that same path cannot be compared against a path with 10 Gbps bandwidth, delay of 1.5 msec, and 0.99 reliability. As a result, forcing a total ordering of path weights may lead to inefficient resource utilization or the inability to satisfy the traffic demands of some flows. In addition, forwarding traffic along paths that incur cycles wastes communication and processing resources.

Designing routing protocols that can compute paths based on quality-of-service (QoS) requirements in a way that such paths are loop-free at every instant is clearly a much needed. component of the Internet routing architecture. However, as the summary of Section II illustrates, the prior work on QoS routing has been rather limited and solving this problem requires a new theoretical framework from which novel routing algorithms can be designed.

Smith and Samson [39] proved that a routing algebra (i.e., a theoretical framework for the design of routing protocols) can be used to design routing protocols that eventually provide *acyclic paths* if and only if "*strict boundedness*" is satisfied, and can be used to design routing protocols that eventually provide *optimal paths* if and only if strict boundedness and "*monotonicity*" are satisfied. These results have also been independently discussed and verified by Sobrinho et al. [40], [41], with boundedness and monotonicity called by different names, including inflationarity and isotonicity, respectively.

Strict boundedness means that the path-weight value of a path P_a resulting from extending another path P_b with a link (a, b) *cannot be better than or equal to* the path-weight value of path $P_b = (a, b)\ P_a$. Monotonicity means that if the path-weight value of a path P_a is better than the path-weight value of another path P_b then extending both paths with links l_a and l_b of equal path-weight values results in the augmented path $l_a\ P_a$ having a path-weight value that *is better than or equal to* the path-weight value of the augmented path $l_b\ P_b$.

In spite of these and similar results, no protocol for QoS routing has been reported to date and proven to converge to acyclic paths or optimal paths using multiple optimality criteria, which motivates the work presented in this paper.

The main contributions of this paper are: (a) Defining a routing algebra for routing on multiple criteria based on route computations that render strict boundedness and monotonicity, (b) introducing the first QoS routing protocol for acyclic routing with path selection based on multiple criteria, and (c) illustrating by examples that the new QoS routing approach can be as efficient as today's shortest-path routing protocols.

Section III uses recent algebraic results to describe the Dominant-Path Routing Algebra (DRA), which formalizes the notion of a *dominant vector* consisting of all path-weight values that describe the performance characteristics of paths required to attain acyclic and optimal routes. Conditions are proven to ensure that a protocol for routing with multiple optimality criteria that satisfies the condition must converge to optimal paths, as well as to be loop-free at every instant.

Section IV applies the results in Section III to show major limitations in recent routing protocols for routing with multiple optimality criteria proposed by Sobrinho and Ferreira [41].

Section V describes the Dominant-Path Routing Information Protocol (DRIP), the first QoS routing protocol that can provide optimal routes based on an arbitrary set of criteria for path selection based on QoS requirements, and without ever incurring routing-table loops. The novelty of DRIP is the use of queries that request paths with path-weight values that are strictly better than the reference weight values they state, and

978-1-6654-6824-4/22/$31.00

updates that add new path-weight values or update or delete old values.

Section VI applies DRIP to traditional shortest-path routing and shortest-and-widest path routing in which paths are selected if they have the shortest length among those that have largest minimum bandwidth. Simulation experiments based on the ns3 simulator are used to illustrate that DRIP is more efficient than even traditional shortest-path routing protocols (RIPv2 [27] and OSPF [30]) and is also more efficient than the protocols proposed by Sobrinho and Ferreira [41] for routing with multiple criteria.

II. Related Work

There has been extensive prior work on algebras applied to routing problems, and they have been called algebras [40], routing metrics [16], path algebras [4], [39]. We choose to use the term *routing algebra*. Gouda and Schneider [16] focused on optimal routing and Smith and Samson [39] extended this work to routing algebras that eventually guarantee loop-free routes, and those that eventually guarantee optimal routes. As we stated in Section 1, DRA is based on these results.

The shortest-path routing algebra [4] permits the computation of optimal-cost paths to destinations. The Distributed Bellman-Ford (DBF) algorithm is used on this algebra and has been used in many routing protocols, including the original ARPANET routing protocol [36], the routing protocol of the DARPA Packet radio network [25], the Routing Information Protocol (RIP) [20], and RIPv2 [27]. However, DBF suffers from the non-convergence problem usually called the counting-to-infinity problem. In practice, routing protocols based on DBF are forced to stop without converging to optimal routes when a predefined maximum-distance value is reached.

Many shortest-path routing protocols have been developed based on the dissemination of partial or complete topology information (e.g., [8], [14], [15], [29], [30]) or the use of complete path information in routing updates (e.g., [24], [42]) to avoid the non-convergence problem of DBF. These routing protocols do not guarantee loop-free routing at every instant, but guarantee convergence by detecting or eventually breaking loops.

Several shortest-path routing protocols (e.g., [33], [34]) have used destination-based sequence numbers to eliminate the convergence problems of DBF and ensure loop-free routing. However, it has been shown that this approach need not provide loop-free routing in all cases when node failures and volatile memory are involved [43].

A number of shortest-path routing approaches have been developed that provide loop-free routes at every instant by requiring nodes to coordinate the updating of routing tables on a multi-hop basis [13], [21], [37]. The most popular of these schemes has been called diffusing computations [13] and is the basis of EIGRP [35].

The techniques for loop detection and avoidance used in shortest-path routing have been applied successfully to multipath routing protocols [31]. Some protocols (e.g., Multipath AODV [28]) use destination-based sequence numbers, others use diffusing computations based on link state information [45], [46] or distance vectors [53], [54], and several use path information [12], [48], [51].

The study of routing subject to multiple performance and policy criteria has its origins in early work on quality-of-service (QoS) routing and approaches that combined multiple link and path constraints into complex cost functions describing performance characteristics of links and paths [5], [7], [17], [18], [22], [32], [44], [47], [49], [52]. At least one approach [26] attains loop-free multi-constrained path selection using diffusing computations [13].

More recent approaches have evolved together with a new focus on routing algebras to address multiple performance and policy criteria. However, this recent work has focused on solutions using complete path information as in BGP or the properties that the routing algebra must satisfy for optimal routing. The recent work by Sobrinho and Ferreira [41] proposes two types of routing protocols for routing on multiple optimality criteria that we analyze in Section IV.

III. Routing with Multiple Criteria

A. Definition of Dominant-Path Routing

We extend the approach introduced by Gouda and Schneider [16] and use the following terminology to define an algebra for routing using multiple criteria: $\mathbb{Z}^+$ is the set of positive integers, N is the set of network nodes (routers and destinations), and E is the set of links in a network. A node in N is denoted by a lower-case letter, and a link between nodes n and m in N is denoted by (n, m). The set of nodes that are immediate neighbor routers of router k is denoted by N^k.

Definition 1: The **Dominant-Path Routing Algebra (DRA)** is the tuple

$$\mathcal{R} = (\Pi, \mathcal{V}, \Omega, \omega_o, \omega_\infty, \varrho, \prec)$$

where:

- Π: The set of node identifiers that is either a subset of $\mathbb{Z}^+$ or a set of alphanumeric strings, and with each identifier assigned uniquely to a router or a destination. The identifier assigned to $n \in N$ is denoted by π^n.
- $\mathcal{V}$: A set of link weights in which each link weight consists of a vector of metric values describing performance characteristics of the link. The weight of the link from router i to router j is denoted by $\nu(i, j)$.
- Ω: The set of path weights, where each such weight consists of a vector of values describing performance- or policy-based characteristics of a path based on the same performance- and policy-based characteristics of links. The weight of the nth path $P_d^k(n)$ from router k to destination d is a denoted by $\omega_d^k(n)$.
- ω_o: The path weight assigned by a destination d for itself.
- ω_∞: The path weight for an unreachable or unknown destination.
- ϱ: A **path-weight function** (PWF) that takes as inputs a link weight $\nu(k, q)$ for the link from a router k to a neighbor q and a path weight $\omega_d^q(j)$ associated with a path $P_d^q(j)$ and returns a path weight $\omega_d^k(n)$ associated with a path $P_d^k(n)$.

- $\prec$: A **weight-induced order relation** defined for any three values $\omega_d^a(i)$, $\omega_d^b(j)$, and $\omega_d^c(k)$ such that the following properties are satisfied:
  1) *Irreflexivity:* $\omega_d^a(i) \not\prec \omega_d^a(i)$
  2) *Transitivity:* For any three different nodes $a \neq b \neq c$, $[(\omega_d^a(i) \prec \omega_d^b(j)) \wedge (\omega_d^b(j) \prec \omega_d^c(k))] \rightarrow (\omega_d^a(i) \prec \omega_d^c(k))$
  3) *Completeness:* $(\omega_d^a(i) \prec \omega_d^b(j)) \vee (\omega_d^b(j) \prec \omega_d^a(i)) \vee (\omega_d^b(j) = \omega_d^a(i)) \vee ([\omega_d^a(i) \not\prec \omega_d^b(j)] \wedge [\omega_d^b(j) \not\prec \omega_d^a(i)] \wedge [\omega_d^b(j) \neq \omega_d^a(i)])$

A path $P_d^k(j)$ consists of a sequence of links, and is also the augmentation of a $P_d^q(i)$ with link (k, q) to node q, i.e., $P_d^k(j) = (k, q)P_d^q(i)$.

A simple example of $\mathcal{V}$ is the set $\mathcal{V}_{min} = \{1, \infty\}$, where a link is assigned a weight equal to 1 if it exists and ∞ if it does not. A simple example of path weights related to $\mathcal{V}_{min}$ is the set Ω_{min} in which $\omega_d^n \in \Omega_{min}$ denotes the hop count of a path from n to d. The PWF ρ_{min} takes a path weight $\omega_d^n \in \Omega_{min}$ and a link weight $\nu(q, n) \in \mathcal{V}_{min}$ as inputs and returns the path weight $\omega_d^n + \nu(q, n)$ representing the hop count along the path P_d^n augmented by link (q, n), i.e., path $P_d^q = (q, n)P_d^n$.

A simple example of $\mathcal{V}$ consisting of vectors is the set of vectors defined by

$$\mathcal{V}_{\delta,\beta} = \{\nu(i, j) \mid ((i, j) \in E) \wedge (\nu(i, j) = [\delta(i, j), \beta(i, j)])\}$$

where $\delta(i, j)$ and and $\beta(i, j)$ are the delay and bandwidth available over link (i, j), respectively. An example of path weights resulting from using $\mathcal{V}_{\delta,\beta}$ is the set $\Omega_{\delta,\beta}$ of path weights $\omega_d^n(i) = [\delta_d^n(i), \beta_d^n(i)]$, where $\delta_d^n(i)$ and $\beta_d^n(i)$ denote the delay and available bandwidth along path $P_d^n(i)$, respectively. A PWF based on $\mathcal{V}_{\delta,\beta}$ and $\Omega_{\delta,\beta}$ is the function $\rho_{\omega(\delta,\beta)}$ that takes a path weight and a link weight as inputs and returns a vector defined by the addition of link delays along a path $P_d^q(i)$ resulting from extending path $P_d^n(j)$ with link (q, n) and the minimum link bandwidth over the same path, i.e., $\omega_d^q(i) = [\delta_d^n(j) + \delta(q, n), Min\{\beta_d^n(j), \beta(q, n)\}]$.

Given that a path-weight value $\omega_d^k(i)$ in DRA consists of a vector of values describing different path characteristics, not all path-weight values can be compared to others. The need to select paths with weights that are better than others while some may not be comparable with one another motivates the following definitions.

Definition 2: **Dominant Path-Weight Value:** A path-weight value $\omega_d^k(i)$ is a dominant weight value over another path-weight value $\omega_d^k(j)$ with $j \neq i$ if $\omega_d^k(i) \prec \omega_d^k(j)$. □

We say that $\omega_d^k(i)$ dominates $\omega_d^k(j)$ if $\omega_d^k(i) \prec \omega_d^k(j)$, and also denote this fact by $\omega_d^b(j) \succ \omega_d^a(i)$ and say that $\omega_d^b(j)$ is dominated by $\omega_d^a(i)$.

Definition 3: **Incomparable Path-Weight Value:** A path-weight value $\omega_d^k(i)$ is incomparable with another path-weight value $\omega_d^k(j)$ with $j \neq i$ if

$$[\omega_d^k(n) \not\prec \omega_d^k(j)] \wedge [\omega_d^k(j) \not\prec \omega_d^k(n)] \wedge [\omega_d^k(j) \neq \omega_d^k(n)]. \ \square$$

Let Ω_d^k be the set of all path-weight values for destination d known at router k, each corresponding to a different path from k to d through one of its neighbors in N^k. The sequence of all path-weight values in Ω_d^k listed in no particular order is denoted by $\{\omega_d^k(i)\}$ with $i = 1, 2, ..., |\Omega_d^k|$.

Definition 4: **Dominant Vector:** The vector of dominant path-weight values known at router k for destination d is denoted by DV_d^k and consists of the subset of path-weight values in Ω_d^k that satisfy the following two conditions:

$$\forall \ (\ \omega_d^k(j) \in DV_d^k \wedge \omega_d^k(n) \in \Omega_d^k - DV_d^k \ ) \tag{1}$$

$$\Big(\ (\omega_d^k(j) \prec \omega_d^k(n)) \vee$$

$$([\omega_d^k(j) \not\prec \omega_d^k(n)] \wedge [\omega_d^k(n) \not\prec \omega_d^k(j)] \wedge [\omega_d^k(n) \neq \omega_d^k(j)]) \Big)$$

$$\forall \ (\ \omega_d^k(i) \in DV_d^k \wedge \omega_d^k(j) \in DV_d^k - \{\omega_d^k(i)\} \ ) \tag{2}$$

$$(\ [\omega_d^k(i) \not\prec \omega_d^k(j)] \wedge [\omega_d^k(j) \not\prec \omega_d^k(i)] \wedge [\omega_d^k(j) \neq \omega_d^k(i)] \ ) \ \square$$

The previous definition simply states that each path-weight value in DV_d^k dominates any other path-weight value in Ω_d^k that is comparable with it, and all path-weight values in DV_d^k are incomparable with one another. A well-designed routing protocol using DRA is such that a router k can select a path $P_d^k(i)$ to a destination d if and only if the path-weight value $\omega_d^k(i)$ associated with $P_d^k(i)$ is in DV_d^k.

B. Characterizing Convergence

A routing protocol based on DRA may be allowed to converge to paths that are acyclic but not optimum according to the set of path-weight values Ω. The following definitions differentiate between simply acyclic and optimal path weights.

Definition 5: **Convergence to Feasible Routes:** A DRA-based routing protocol converges to paths with feasible path-weight values if, for any destination d, any router k: (a) obtains a set of dominant path-weight values DV_d^k corresponding to acyclic paths or sets $\omega_d^k = \omega_\infty$ within a finite time after network changes stop occurring, and (b) does not change the value of any path weight subsequently. □

Definition 6: **Convergence to Optimal Routes:** A DRA-based routing protocol converges to paths with optimum path-weight values if, for any destination d, any router k: (a) obtains a set of path weights that equals the dominant vector of path weights for k to destination d within a finite time after network changes stop occurring, and (b) does not change the value of any path weight subsequently □

Definition 7: **Loop Freedom:** A routing protocol is acyclic (loop-free) if, for any destination d, all the paths implied by the routing information maintained by routers define acyclic paths at every instant. □

The next hop along the nth path $P_d^k(n)$ from router k to destination d is denoted by $s_d^k(n)$. Hence, path $P_d^k(n)$ consists of the concatenation of the link $(k, s_d^k(n))$ with a path $P_d^{s_d^k(n)}(m)$ offered by $s_d^k(n)$ to k, i.e., $P_d^k(n) = (k, s_d^k(n))P_d^{s_d^k(n)}(m)$.

The path-weight value reported by $s_d^k(n)$ to router k for destination d is denoted by $\omega_{ds_d^k(n)}^k(n)$.

The key objective of any routing protocol is to establish paths to destinations of interest such that traversing each hop along any path gets closer to the destination of the path, which requires the next router along the path to be closer to the destination than the previous router. The following definition formalizes the intuition that, independently of the type of partial ordering defined among path-weight values in a routing algebra, if router k uses router q as its next hop along a path to destination d the path from q to d must be better than the path from k to d that has q as the next hop.

Definition 8: **Total Order along Simple Paths:** The total ordering along a simple path from router k to a destination d is given by

$$\mathcal{T} : \omega^k_{ds^k_d(n)}(n) \prec \omega^k_d(n) \qquad \square \tag{3}$$

C. Conditions for Convergence and Optimality

The following nomenclature is used in theorems that establish the conditions for the existence of routing protocols based on DRA that converge to acyclic (feasible) paths or optimum paths, and that can be acyclic at every instant, even before converging.

$\mathcal{F}$ denotes the fact that a routing protocol convergences to paths with feasible path-weight values.

$\mathcal{O}$ denotes the fact that a routing protocol converges to paths with optimum path-weight values.

$\mathcal{A}$ denotes the fact that a routing protocol is acyclic.

$\mathcal{T}^*$ denotes the fact that a routing protocol satisfies $\mathcal{T}$ at every instant.

$\mathcal{D}$ denotes the fact that a routing protocol is such that each router knows within a finite time its complete and valid dominant vector for each destination of interest.

Theorem 1: If a routing protocol based on DRA converges to feasible path-weight values, the total ordering constraint $\mathcal{T}$ must be satisfied by every router that must maintain paths to any destination d within a finite time after network changes stop occurring.

Proof: The proof needs to show that $\mathcal{F} \to \mathcal{T}$ is true. The argument of the proof is by contradiction, i.e., by showing that $\mathcal{F} \wedge \neg\mathcal{T}$ is a contradiction.

Assume that a routing protocol converges at time t_0 but $\mathcal{T}$ is not satisfied by at least one node k at that time. From the definition of convergence, no router can change the path-weight value of any path after time t_0 and no router can transmit a signaling message to update a path-weight value. Hence, router k cannot change the path-weight value $\omega^k_d(n)$ of a path $P^k_d(n)$ that does not satisfy $\mathcal{T}$ after time t_0.

Let $q = s^k_d(n)$ be the next hop along path $P^k_d(n)$; router k used $\omega^k_{dq}(n)$ to select q as the next hop along $P^k_d(n)$, and $\omega^k_{dq}(n)$ corresponds to a path-weight value $\omega^q_d(m)$ communicated to k by q and corresponding to the mth path $P^q_d(m)$ from q to d. Furthermore, $\omega^q_d(m) = \omega^k_{dq}(n)$ cannot change after time t_0. This allows k to use $q = s^k_d(n)$ even if it true that $k = s^q_d(m)$, i.e., k is the next hop for q along path $P^q_d(m)$. This is a contradiction, because then P^k_d cannot be a simple path. ■

Corollary 1: If a routing protocol based on DRA converges to optimal path-weight values, the total ordering constraint $\mathcal{T}$ must be satisfied by every router that must maintain paths to any destination d within a finite time after network changes stop occurring.

Proof: The proof needs to show that $\mathcal{O} \to \mathcal{T}$ is true, and the argument of the proof is by contradiction. The structure of proof is much the same as in the proof of Theorem 1 with the only difference that routers converge to optimal values at time t_0 rather than just feasible values. ■

Theorem 2: A routing protocol based on DRA is acyclic **if and only if** the total ordering $\mathcal{T}$ is satisfied at every instant by every router that must maintain paths to any destination d.

Proof: The proof needs to show that $\mathcal{A} \leftrightarrow \mathcal{T}^*$ is true. The argument of the proof consists of two parts: showing that $\mathcal{T}^* \to \mathcal{A}$ is true, and showing that $\mathcal{A} \to \mathcal{T}^*$ is true.

($\mathcal{T}^* \to \mathcal{A}$): The proof proceeds by contradiction. For this purpose, assume that $\mathcal{T}^*$ is true but $\mathcal{A}$ is false because a set of routers $V = \{v_1, v_2, ..., v_h, v_1\}$ with $d \notin V$ creates a routing-table loop L of h hops at some point in time. Without loss of generality, label the path used by each such router $v_i \in V$ to create loop L as $P^{v_i}_d(1)$. Loop L is thus created when each router v_i sets $s^{v_i}_d(1) = v_{i+1}$ for $1 \le i \le h-1$ and $s^{v_h}_d(1) = v_1$. This means that $\omega^{v_{i+1}}_d(1) \prec \omega^{v_i}_d(1)$ for $1 \le i \le h-1$ and $\omega^{v_1}_d(1) \prec \omega^{v_h}_d(1)$. This is a contradiction, because it implies that $\omega^{v_i}_d(1) \prec \omega^{v_i}_d(1)$ for $1 \le i \le h-1$ and $\omega^{v_h}_d(1) \prec \omega^{v_h}_d(1)$.

($\mathcal{A} \to \mathcal{T}^*$): The proof also proceeds by contradiction Assume that $\mathcal{A}$ is true and also assume that $\mathcal{T}^*$ is not true because $\mathcal{T}$ is not satisfied by at least one router k with respect to a destination d at some time t_L. Without loss of generality, consider a scenario in which routers p and k have single simple paths to destination d such that $s^p_d(1) = k$, $s^k_d(1) = q \neq p$, and $\omega^k_d(1) \prec \omega^p_d(1)$ at some time $t < t_L$. Suppose that router k must change its path to d at time t_L and its only available neighbor is p. Router k can make $s^k_d = p$, which is a contradiction because $P^k_d(1) = (k,p)P^p_d(1)$ at time t_L is not an acyclic path and hence $\mathcal{A}$ cannot be true. ■

Theorem 3: A routing protocol based on DRA with a set of path-weight values Ω in which optimal values exist converges to those optimal values at time $t_\mathcal{O}$ **if and only if** each router has a complete and valid dominant vector for each destination d at time $t_\mathcal{O}$.

Proof: The proof needs to show that $\mathcal{O} \leftrightarrow \mathcal{D}$ is true.

($\mathcal{O} \to \mathcal{D}$): This part of proof is by contradiction. Assume that a routing protocol converges to optimal values at time $t_\mathcal{O}$ and for the sake of contradiction assume that at least one router s does not have the complete and valid dominant vector DV^s_d for destination d at time $t_\mathcal{O}$. There are only two possible cases to consider for router s at time $t_\mathcal{O}$ based on this assumption: (a) There is at least one path-weight value $\breve{\omega}^s_d$ in DV^s_d that should not be in the set, or (b) there is at least one path-weight value $\dot{\omega}^s_d$ that is not in DV^s_d but should be in it.

Assume that router s stores $\breve{\omega}^s_d \in DV^s_d$ at time $t_\mathcal{O}$. Given that $\breve{\omega}^s_d$ should not be in DV^s_d at time $t_\mathcal{O}$ because it is no longer valid, it follows from Definition 4 that the following is true at time $t_\mathcal{O}$: $\forall \omega^s_d(i) \in DV^s_d - \{\breve{\omega}^s_d\}$ ($\omega^s_d(i) \prec \breve{\omega}^s_d$). This is a contradiction, because then router s could use $\breve{\omega}^s_d$ to forward data to d along a path that is not optimal at time $t_\mathcal{O}$.

Assume that $\dot{\omega}^s_d \notin DV^s_d$ at time $t_\mathcal{O}$. Because by assumption $\dot{\omega}^s_d$ is in DV^s_d at time $t_\mathcal{O}$, it follows from Definition 4 that the following two statements are true at time $t_\mathcal{O}$:

$$\forall\ \omega^s_d(i) \in DV^s_d - \{\dot{\omega}^s_d\}$$

$$\left[\ [\dot{\omega_d^s} \not\prec \omega_d^s(i)] \wedge [\omega_d^s(i) \not\prec \dot{\omega_d^s}] \wedge [\omega_d^s(i) \neq \dot{\omega_d^s}]\ \right]$$
$$\forall \omega_d^s(j) \in \Omega_d^s - \{\dot{\omega_d^s}\} \left[\ \ (\dot{\omega_d^s} \prec \omega_d^s(j)) \vee \right.$$
$$\left. (\ [\dot{\omega_d^s} \not\prec \omega_d^s(j)] \wedge [\omega_d^s(j) \not\prec \dot{\omega_d^s}] \wedge [\omega_d^s(j) \neq \dot{\omega_d^s}]\ )\ \ \right]$$

This is also a contradiction, because it implies that router s can forward data along a path that has a path-weight value that is worse than the path associated with path-weight value $\dot{\omega_d^s}$, which is not optimum.

($\mathcal{D} \rightarrow \mathcal{O}$): This part of the the proof is also by contradiction. Assume that all routers maintain complete and valid dominant vectors for destination d at time $t_\mathcal{O}$. For the sake of contradiction assume that at least one router s has converged to a route for destination d that is not optimal at time $t_\mathcal{O}$. Let the path-weight value associated with that route be ω_d^{*s}. Given that the routing protocol is based on DRA, $\omega_d^{*s} \in DV_d^s$ and hence,

$$\forall \omega_d^s(j) \in \Omega_d^s - \{\omega_d^{*s}\}\ \left[\ (\omega_d^{*s} \prec \omega_d^s(j)) \vee \right.$$
$$\left. ([\omega_d^{*s} \not\prec \omega_d^s(j)] \wedge [\omega_d^s(j) \not\prec \omega_d^{*s}] \wedge [\omega_d^s(j) \neq \omega_d^{*s}])\ \right]$$

However, this is a contradiction to ω_d^{*s} not being an optimal path-weight value. ■

IV. Limitations of Recent Protocols for Routing with Multiple Criteria

Sobrinho and Ferreira proposed protocols for routing on multiple optimality criteria [41]. These routing protocols are called Non-Restarting Vectoring Protocol (denoted by NRVP), and Restarting Vectoring Protocol (denoted by RVP). We prove that NRVP is not correct and show that RVP incurs considerable delays and communication overhead. We assume in our analysis that all protocol messages are exchanged correctly among routers, but link and router failures may occur.

A. Analysis of NRVP

NRVP suffers from non-convergence because its signaling does not satisfy the condition stated in Theorem 1, and suffers from looping because its signaling does not satisfy the condition stated in Theorem 2.

The proof of termination of NRVP (Appendix A in [41]) is incorrect because it does not address the way in which the signaling of NRVP really works and assumes a finite set of path attributes. The signaling in NRVP simply makes a router find a new set of dominant path attributes after receiving updates from neighbors and report that to its neighbors (see Algorithm 1 in [41]). As a result, NRVP suffers from looping and non-termination problems after link failures, node failures, or some changes in path-weight values.

Figure 1 illustrates looping and non-convergence in NRVP with a simple example consisting of a five-node network. The delay δ and bandwidth β over a link are indicated by (δ, β) next to each link. The dominant path-weight values for destination d consisting of vectors with path delay $\sum \delta$ equal to the sum of the bandwidth of link in the path and path bandwidth $Min(\beta)$ equal to the minimum link bandwidth along the path. Dominant path-weight values are indicated by $[\sum \delta, Min(\beta)]$ next to each node followed by the next hop of the path with the dominant path-weight value. Arrowheads in links indicate the next hops to destination d. Updates are indicated by $U[\sum \delta, Min(\beta)]$, and routing state and signaling involved in looping are shown in red.

Figure 1(a) shows the initial state at the routers just before link (b, d) fails, and Figures 1(b) to 1(d) illustrate the updates made to the dominant path-weight values by the routers. Dashed arrows indicate the transmission of update messages to their neighbors. For brevity, Figures 1(b) to 1(d) each show two steps in the transmission of updates without showing the link metrics, which are the same as in Figure 1(a).

As the example illustrates, routers a and b become part of a routing-table loop while seeking to establish paths associated with path-weight values with a minimum bandwidth of 10, which are incomparable with path-weight values with a minimum bandwidth of 1, and the loop persists independently of other incomparable routes.

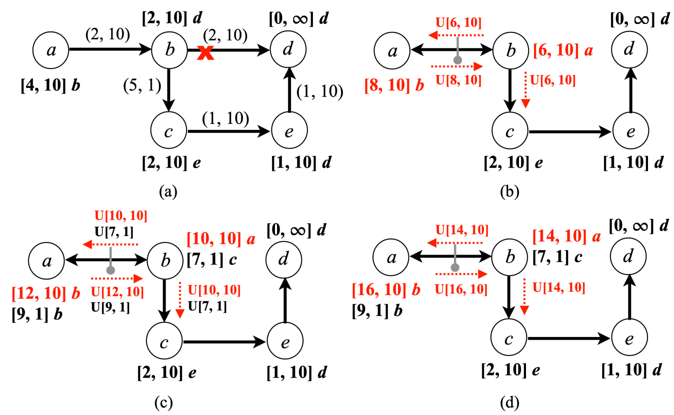

Figure 1: Looping and non-convergence in NRVP

This example illustrates that non-convergence in NRVP may occur even when physical paths to destinations exist. An external measure is required to force NRVP to *stop without convergence*, which is exactly what is done in protocols like RIP that use DBF. A maximum hop count set to 15 is used in [41] to analyze the performance of NRVP by simulation, which renders results that do not prove correct termination.

B. Analysis of RVP

The signaling in RVP uses destination sequence numbers to keep the destination in control of the partial ordering of path-weight to itself. RVP requires routers to trust only attributes associated with a higher sequence number as the one currently stored for a destination, or dominant path attributes with the same sequence number (see Algorithm 2 in [41]). RVP avoids the non-convergence problems found in NRVP as long as sequence numbers are correct; however, like all prior routing protocols based on destination sequence numbers, it suffers from delayed convergence, and does not work correctly if sequence numbering is not correct as a result of the recycling of sequence numbers drawn from a finite sequence-number space, or the rebooting of routers that do not use non-volatile memory to maintain all the sequence numbers used by the protocol.

The reader is referred to prior work [3], [43] for detailed discussions of the correctness problems of any routing protocol based on destination sequence numbers. The focus of our analysis of RVP is on the long delays incurred by routers to obtain new valid paths and the large amount of information that routers should communicate to ensure correctness.

Figure 2 illustrates the delayed convergence problems in RVP when sequence numbers are correct using the same example of Figure 1. Sequenced dominant path-weight values are indicated by $[n : \sum \delta, Min(\beta)]$ next to each node followed by the next hop of the path with the dominant path-weight value, where n denotes the value of the sequence number originated by destination d. Updates are denoted by $U[n : \sum \delta, Min(\beta)]$ and routing state associated with delayed convergence is indicated in red. In the example shown in Figure 2, router b loses its single dominant path to destination d when link (b, d) fails and must wait to receive updates with a sequence number of $n + 1$ to correct its state.

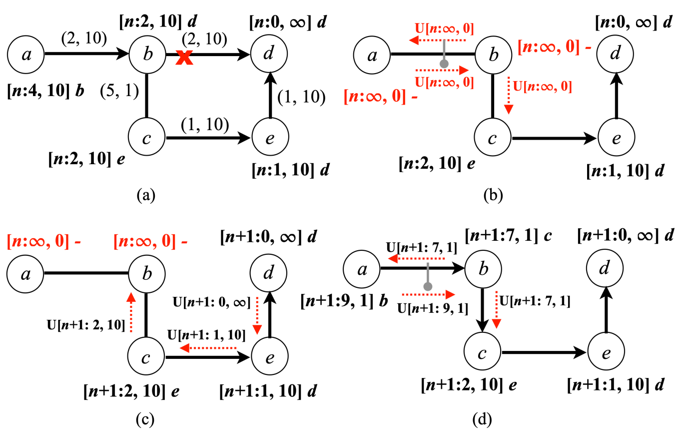

Figure 2: Delayed convergence in RVP

This example illustrates that routers executing RVP may not obtain valid paths corresponding to new dominant path-weight values for some time, until the destination disseminates updates with new sequence numbers. In addition, when a router receives an update for the destination with a more recent sequence number, RVP requires the router to communicate its *entire dominant vector* for a destination to work correctly, not just incremental updates. This results in considerable more signaling overhead compared to shortest-path routing.

V. DRIP: Dominant-Path Routing Information Protocol

Our description of DRIP assumes that routers send routing messages reliably after waiting for short or long time intervals.

The objective of DRIP is to maintain acyclic paths to destinations based on multiple criteria for path selection. DRIP establishes a partial ordering among path-weight values with respect to a given destination based on DRA, and each router also maintains a total ordering at every instant between itself and the next hop along any specific path to a destination, which enforces $\mathcal{T}$. The signaling in DRIP is based entirely on changes to the dominant vectors of path-weight values maintained by routers for different destinations. It can be viewed as a loop-free extrapolation of the signaling used in traditional distance-vector protocols for shortest-path routing.

A. Information Exchanged

Routers exchange routing messages reliably among one another to update their routing information. A routing message from router k is denoted by M^k and contains its node identifier π^k, a list of updates, and a list of queries.

A **query** from router k is a tuple $(o, \pi^d, \pi^s, p^k_d, n^k_d, r^s_d)$, where the operation code o equals Q to indicate that the entry is a query; π^d is the node identifier of a destination d; π^s is the node identifier of the source s of the query; p^k_d is the node identifier of the prior router forwarding the query; n^k_d is the node identifier of a router that should forward the query; and r^s_d is the reference weight value set by the source of the query.

Stating $n^k_d = d$ in a query indicates that all routers receiving the query should process it.

An **update** from router k is a tuple $(o, \pi^d, \pi^s, n^k_d, r^s_d, \omega^k_d)$, where the operation code o equals A if the update modifies or adds a path-weight value, and equals D if the update deletes a path-weight value; π^d is the node identifier of a destination d; π^s is the node identifier of the source s of the query that prompted the update n^k_d is the identifier of a router that should send the update towards π^s; r^s_d is a reference weight value for destination d stated by the source of the query that prompted the update; and ω^k_d is the weight of a path from router k for destination d. An update with $\pi^d = \pi^s$ states that the destination started the update proactively. Stating $n^k_d = \pi^d$ in an update indicates that all routers receiving the update are allowed to send their own updates towards source π^s, or to all their neighbors if $\pi^d = \pi^s$. Stating $n^k_d = 0$ with $\pi^d = \pi^s$ in an update indicates that the update is intended only for immediate neighbors as a "hello" to refresh the presence of node k.

B. Information Maintained

Each router k maintains a Link-Weight Table (LWT^k), a Neighbor Table (NT^k), a Routing Table (RT^k), and knows its own identifier π^k.

LWT^k lists an entry for each link to a known neighbor router $n \in N^k$. The entry for link (k, n) in LWT^k states: (a) The weight $\nu(k, n)$ of the link, and (b) a lifetime LT^k_n for the neighbor entry of node n.

NT^k lists the dominant vectors reported by each neighbor for each destination of interest. The entry in NT^k for destination d at router k is denoted by $NT^k(d)$ and specifies for each neighbor $p \in N^k$: (a) The identifier of a destination d (π^d), and (b) the dominant vector for destination d reported by p (denoted by DV^k_{dp}).

The weight of the jth path in DV^k_{dp} is denoted by $\omega^k_{dp}(j)$.

If a neighbor q has not reported any path-weight value for d to router k, then router k assumes that $\omega^k_{dq} = \omega_\infty$.

RT^k lists an entry for each destination d for which the router must maintain routing information. The entry in RT^k for destination d is denoted by $RT^k(d)$ and states:

• π^d: The identifier of destination d

• DV_d^k: The dominant vector for destination d, which contains the path weight of each of its selected paths for the destination.

The ith entry in DV_d^k $(1 \leq i \leq |DV_d^k|)$ consists of the tuple $[\omega_d^k(i), DS_d^k(i)]$, where $\omega_d^k(i)$ is a path weight that is incomparable with any other path weight in DV_d^k, and $DS_d^k(i)$ is the *dominating successor set* for $\omega_d^k(i)$ defined by:

$$DS_d^k(i) = \{\pi^q \mid (q \in N^k) \wedge (\exists\ \omega_{dq}^k(j) \prec \omega_d^k(i))\}$$

The dominating successor set at router k for destination d is denoted by DS_d^k and equals:

$$DS_d^k = \bigcup_{i=1}^{|DV_d^k|} DS_d^k(i)$$

C. Updating Routing Information

Initialization: Router k is initialized to store ω_o for itself, and the maximum path-weight ω_∞ is assumed for any other destination. Entries for other destinations are added as updates are received from neighbors. Once a router is initialized, it transmits routing message with an add update for itself according to its update timers.

Timing of Routing Messages: If router k needs to send a routing message with updates or queries, it does so after a minimum amount of time t_{min} has elapsed from the time it sent its prior routing message.

In the absence of queries or updates needed to reflect changes to routing tables, a router sends a message with a "hello" update no later than t_{max} seconds from the time it sent its last message, where t_{max} is shorter than the maximum lifetime of a LWT entry (LT). For this purpose, router k sets UT^k equal to t_{max} after sending a routing message, and sets UT^k equal to t_{min} after preparing updates or queries to be sent in response to input events.

A "hello" update consists of the tuple $(A, \pi^k, \pi^k, 0, \omega_o, \omega_o)$ and simply updates the lifetime entries maintained for itself by its neighbors.

Updating Local Information: Router k updates LWT^k when the weight $\nu(k, q)$ of an outgoing link to q changes, and updates NT^k and RT^k when any input event occurs, such as when an adjacent outgoing link changes its weight, an immediate neighbor router fails to send updates before the lifetime for its LWT entry expires, or a routing message M^q is received from a neighbor q.

To update its routing table, router k uses the path-weight function ρ to compute new routing-metric values for a given destination d after the occurrence of an input event. The router adds a *delete update* to its routing message if a path-weight value is no longer valid and is incomparable to its remaining path-weight values. A router includes *add updates* to its routing message to report updates to path-weight values that could be compared to prior ones, or to report new incomparable path-weight values.

A router can select a neighbor router as a next hop along any new or existing path only if $\mathcal{T}$ is satisfied along that path, in which case the router sends an add update as needed. Otherwise, if $\mathcal{T}$ cannot be satisfied for a path with a given path-weight value, the router sends a query using that value as the reference weight of the query. To ensure that dominant vectors contain only valid path-weight values, a router that eliminates a path-weight value from its own dominant vector sends a delete update to its neighbors informing them that the path-weight value was deleted.

A router k that has a path $P_d^k(n)$ to destination d with $\omega_d^k(n) \prec \omega_\infty$ and needs to update or change that path uses $\omega_d^k(n)$ as a reference point to search for an immediate neighbor or a remote router that can provide a routing-metric value with which $\mathcal{T}$ is satisfied. If router k has a neighbor router q with $\omega_d^q(m) \prec \omega_d^k(n)$, then k can use q as its next hop in a new acyclic path to d, and router k just informs its neighbors about the update made to its dominant vector. Otherwise, k must start a search for a remote router that satisfies $\mathcal{T}$ by sending a query to the rest of the network stating a reference weight $r_d^k = \omega_d^k(n)$.

Processing Queries and Updates: A router j receiving a query stating a reference weight r_d^k can respond to the query with an update if it has a path-weight value $\omega_d^j(m) \prec r_d^k$ and forwards the query otherwise. The next hop for a forwarded query is any neighbor along the set of fast preferred paths to the destination stated in the query, denoted by $DS_d^k(1)$.

If router j is able to respond to a query from router k, its update propagates back to router k and the relays in between update their path-weight values as they propagate the update towards k, so that $\mathcal{T}$ is satisfied by every router at every instant.

On the other hand, if router k does not have a current path-weight value better than ω_∞ and must find a path to d, then it does not have any point of reference for its search. Therefore, router k must send a query stating a reference weight $r_d^k = \omega_o$ so that only the destination d can respond with an update that can be trusted.

D. Example of DRIP Operation

Figure 3 illustrates the fast convergence of DRIP using the same example of Figures 1 and 2.

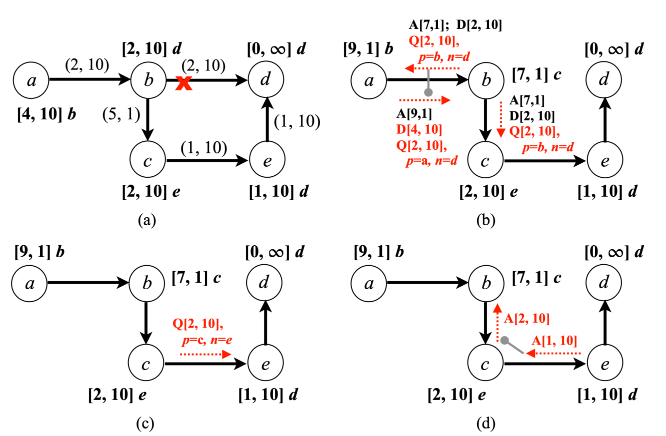

Figure 3: Example of DRIP operation

Queries are highlighted in red and denoted by $Q[\sum \delta, Min(\beta),\ p = x, n = y]$, which indicates the reference weight of the query, followed by the previous node p that forwarded the query and the next node n that should process the query. Add and delete updates are denoted by $A[\sum \delta, Min(\beta)]$ and $D[\sum \delta, Min(\beta)]$, respectively. Updates sent back in response to the query originated by node b are highlighted in red.

The failure of link (b, d) Figure 3(a) forces router b to send a query to all its neighbors to start a search for a router that has a better routing-metric value than $[2, 10]$ (Figure 3(b)). In addition, router b sends an update to add weight $[7, 1]$ and an update to delete weight $[2, 10]$. Router a retransmits the query to its neighbor b as a result of the query from b, and also sends an update to add weight $[9, 1]$ and an update to delete weight $[4, 10]$ as a result of the updates received from b (Figure 3(b)). Given that $[2, 10] \not\prec [2, 10]$, router c must forward the query towards d, and does this by making the next hop for the query it sends equal to e (Figure 3(c)). Router e sends back an add update in response to the query because $[1, 10] \prec [2, 10]$, and this makes c do the same (Figure 3(d)). As this example illustrates, no loop ever occurs in DRIP and routers quickly converge to new valid routing-metric values and delete invalid values even if they are incomparable with new ones.

E. DRIP Correctness

The proofs that DRIP is always acyclic and that it converges to feasible or optimal paths follow directly from Theorems 2 and 3. We provide only a sketch of the proof of Theorem 5 due to space limitations.

Theorem 4: DRIP is acyclic for any destination of interest.

Proof: Let $\mathcal{L}$ denote the fact that the signaling of DRIP is executed correctly at every instant by every router of the network. Given the result of Theorem 2, the proof needs to show that $\mathcal{L} \rightarrow \mathcal{T}^*$ is true.

According to the correct operation of DRIP, if router k has no finite routing-metric value for destination d then its routing metric value for d is $[\omega_\infty, \pi^k]$ and it does not have any next hop along a path to destination d. In addition, if router k selects router $n \in N^k$ as its next hop along a path $P^k_d(i)$ to d with a routing metric value $\mu^k_d(i) \prec [\omega_\infty, \pi^k]$ then $\mu^k_{dn}(j) \prec_\mu \mu^k_d(i)$, where $\mu^k_{dn}(j)$ to k is a routing metric reported by n to k.

Assuming that $\mathcal{L}$ is true implies that it is true at any time time t_L, and from the previous argument it must be true that the following is true at time t_L:

$$\mu^k_{dn}(j) \prec_\mu \mu^k_d(i) \quad (4)$$

For the the sake of contradiction also assume that $\mathcal{T}^*$ is not satisfied by at least router k for destination d. This implies that router k does not satisfy $\mathcal{T}$ for destination d. The definition of $\mathcal{T}$ implies that the following is true if $\neg\mathcal{T}$ is true at time t_L: $\mu^k_{dn}(j) \not\prec_\mu \mu^k_d(i)$, which is a contradiction to Eq. (4) and therefore the theorem is true. ■

Theorem 5: DRIP converges to feasible routing-metric values for any given destination of interest within a finite time after network changes stop occurring in the network.

Sketch of Proof: From Theorem 3, the proof needs to show that the exchange of dominant vectors for a given destination of interest among routers results in optimal routing-metric values at every router within a finite time after network changes stop occurring. The proof proceeds by contradiction using the fact that DRIP is loop-free at every instant and the assumption that routers communicate their dominant vectors reliably. ■

VI. PERFORMANCE COMPARISON

We apply DRIP to shortest-path routing and widest-and-shortest path routing to illustrate its efficiency compared to traditional routing protocols based on shortest-path routing and NRVP, which we discussed in Section IV. We simulate DRIP, NRVP, RIPv2, and OSPF using the network simulator ns3. The purpose of the simulation experiments is simply to verify that DRIP has no loops, that single-metric DRIP results in faster convergence times and reduced signaling overhead than RIPv2 or OSPF, and that routing on multiple metrics need not result in major increases in the signaling overhead or unacceptably long convergence times. We used several real-world topologies for the simulations (see Table I), selecting topologies with potential routing loops and multiple possible paths to each destination. We implemented DRIP in ns3 by modifying the ns3 implementation of RIPv2 publicly available. We implemented NRVP in ns3 based on the description in [41] by modifying the ns3 implementation of RIPv2 to account for multiple parameters being used in a path metric. We used the ns3/Quagga implementations of RIPv2 and OSPF without modification. The signaling of DRIP is such that its convergence time is not a function of the number of parameters used to characterize links, as long as larger routing messages can be transmitted to communicate updates to the dominant vectors. Similarly, the convergence of NRVP is subject to the maximum count of 15 to stop route computations in case of non-convergence. Therefore, the convergence time of multi-metric RIPv2 (our implementation of Sobrinho and Ferreira's NRVP) and single-metric RIPv2 are similar to each other.

Parameter	Value
Node Count: UK Backbone Toplology [10]	8
Node Count: Custom Topology	11
Node Count: Abilene Topology [2]	12
Node Count: NSFNET-T1 Topology [53]	13
Node Count: ATT N. America Topology [2]	25
Simulation Time	1000 sec
Hello Interval	2 sec
Update Interval in DRIP	2 sec

Table I: Simulation parameters

RIPv2 uses only hop count to determine routing cost. By contrast, DRIP can use any path metrics that conform to the constraints of monotonicity (isotonicity) and strict boundedness (inflation) described in [41]. For the DRIP simulation, we used a link cost of 1 per link, and introduced bandwidth as a second metric, where each link had a bandwidth of either 1 Mbps or 10 Mbps. When comparing paths to a destination, if path a had both higher bandwidth and lower hop count than path b, then the node routed packets to path a. If the bandwidth of path a was higher, but the hop count of path b was higher, i.e., the two weights of the paths were incomparable, packets were routed randomly either to maximize bandwidth or minimize hop count. Signaling overhead was measured as the total number of bytes the routing messages incurred during the simulation. Nodes were connected in the simulation setup via a shared link with a two second delay and a 5 Mbps data rate. The nodes shared links by running Carrier Sense Multiple Access (CSMA) links across the shared medium.

Figure 4 compares the performance of OSPF, RIPv2, NRVP, single-metric DRIP and dual-metric DRIP in terms of signaling overhead. DRIP incurs significantly less signaling overhead than OSPF and less signaling overhead than NRVP and RIPv2. This was expected, because the packet size is much smaller for DRIP than RIPv2, and the amount of information that must be sent is smaller in DRIP than in OSPF and NRVP as the number of links increases.

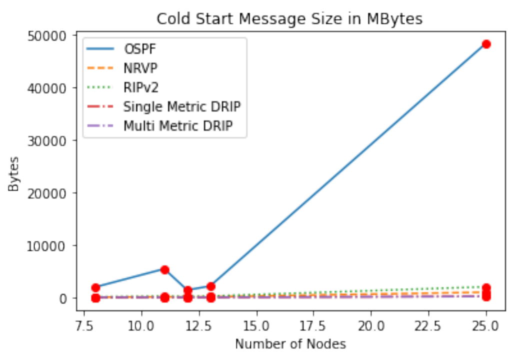

Figure 4: Messages sent by each protocol in bytes

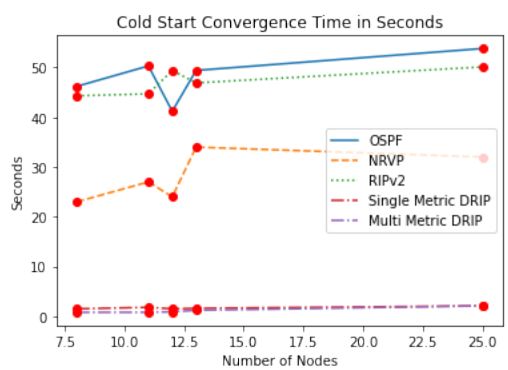

Figure 5: Convergence time of routing protocols

Figure 5 shows the convergence time for each protocol vs. the number of nodes in the network. Updates occur more frequently in DRIP than the other protocols, which enables convergence to occur much faster. NRVP and multi-metric DRIP converge much faster than RIPv2 and single-metric DRIP. This is due to the fewer number of routes available with higher bandwidth. We did not test node failure specifically on nodes that would cause the higher bandwidth route to fail, so did not test node recovery specifically to re-achieve higher bandwidth routes after failure.

In the case of the 25-node network, the convergence time for multi-metric routing was significantly faster than expected. This is due to the relative lack of high bandwidth routes as compared to routes with short hop counts. In addition, a node in this topology has an average degree of four, while the highest-degree node had eight links. Dual-metric DRIP had very similar convergence times to single-metric DRIP. However, this may be a result of the rapid convergence times of DRIP as a whole. Overall, the results indicate that the added flexibility of multiple metrics does not come at a cost of convergence delay for DRIP.

We also simulated a node failing at 40 seconds into the simulation for RIPv2, OSPF, single-metric DRIP. Figure 6 shows the signaling overhead to recover from node failure for each protocol, and Figure 7 shows the convergence time for single-metric RIPv2, OSPF and DRIP. The results show that the time for the network to recover after a node failure is much higher for RIPv2 than for OSPF, which is expected due to the known counting-to-infinity problem of RIPv2. The convergence time for DRIP is minimal, which is due to the fact that DRIP is acyclic and its timers can be much shorter than in OSPF and RIPv2.

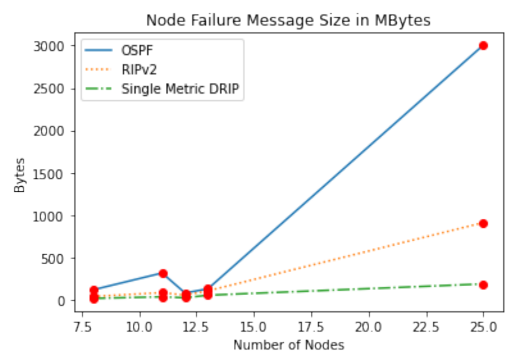

Figure 6: Signaling overhead to reconfigure after a node failure

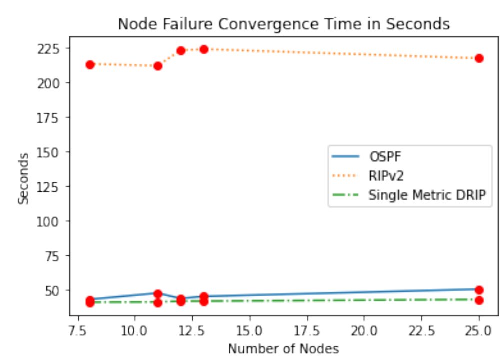

Figure 7: Time to reconfigure after a node failure

VII. Conclusion

We proved necessary and sufficient conditions for protocols for routing with multiple performance criteria used for path selection to be loop-free at every instant and to converge to optimal values. We showed that recent protocols by Sobrinho and Ferreira [41] for routing on multiple optimality criteria suffer from non-convergence or delayed convergence, because they do not satisfy the necessary and sufficient conditions we proved. To address this, we introduced DRIP, the first protocol for routing with multiple criteria that is loop-free at every instant and is guaranteed to converge to valid (feasible or optimal) paths. DRIP relies on a local or global search for paths that satisfy the necessary and sufficient condition for loop-freedom using queries that state reference weight values.

Simulation experiments using ns3 illustrate that DRIP is more efficient than even traditional shortest-path routing protocols by comparing it with OSPF and RIPv2, and that its signaling is equally efficient when multiple criteria are used for path selection. Additional analysis through analytical models (e.g., [23], [50]) and simulation is needed to fully understand the performance advantages of DRIP.

REFERENCES

[1] P. Baran, "On Distributed Communications: I. Introduction to Distributed Communications Networks," Memorandum RM-3420-PR, RAND Corporation, Aug. 1964.

[2] The University of Adelaide, "The Internet Topology Zoo," The Internet Topology Zoo, 2013. http://www.toplogy-zoo.org/index.html.

[3] K. Bhargavan et al., "Formal Verification of Standards for Distance Vector Routing Protocols," *Journal of the ACM*, July 2002.

[4] B.A. Carre, "An Algebra for Network Routing Problems," *J. Inst. Maths Applics.* , 1971.

[5] D. Cavendish and M. Gerla, "Internet QoS Routing Using the Bellman-Ford Algorithm," *Proc. IFIP HPN '98*, Sept. 1998.

[6] G.P. Chandranmenon and G. Varghese, "Trading Packet Headers for Packet Processing," *IEEE/ACM Trans. Networking*, 1996.

[7] S. Chen and K. Nahrstedt, "An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions," *IEEE Network*, ,Nov. 1998

[8] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," *RFC 3626*, 2003.

[9] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*, Morgan Kaufmann, 2000.

[10] M. Dodge, "An Atlas of Cyberspaces-ISP Backbone Maps," Maps of Internet Service Provider (ISP) and Internet Backbone Networks, 2004.

[11] A. Flavel and M. Roughan, "Stable and Flexible iBGP," *Proc. ACM SIGCOMM '09*, Aug. 2009.

[12] Igor Ganichev et al., " YAMR: Yet Another Multipath Routing Protocol," *ACM SIGCOMM CCR*, Oct. 2010.

[13] J.J. Garcia-Luna-Aceves, "Loop-Free Routing Using Diffusing Computations," *IEEE/ACM Trans. Networking*, 1993.

[14] J.J. Garcia-Luna-Aceves and J. Behrens, "Distributed, Scalable Routing based on Vectors of Link States," *IEEE JSAC*, Oct. 1995.

[15] J. J. Garcia-Luna-Aceves and M. Spohn, "Scalable Link-State Internet Routing," *Proc. IEEE ICNP '98*, Oct. 1998.

[16] M. Gouda and M. Schneider, "Maximizable Routing Metrics," *IEEE/ACM Trans. Networking*, Aug. 2003.

[17] R. Guerin, A. Orda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions," *Proc. IEEE Globecom '97*, Nov. 1997.

[18] R.A. Guerin and A.Orda, "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms," *IEEE/ACM Trans. on Networking*, June 1999.

[19] E. Harzheim, *Ordered Sets*, Springer, 2005.

[20] C.L. Hedrick, "Routing Information Protocol" *RFC 1058*, 1988.

[21] J.M. Jaffe and F.M. Moss, "A Responsive Routing Algorithm for Computer Networks," *IEEE Trans. on Communications*, July 1982.

[22] J.M. Jaffe, "Algorithms for Finding Paths with Multiple Constraints," *Networks*, 1984.

[23] M. .J. Johnson, "Updating Routing Tables after Resource Failure in a Distributed Computer Network," *Networks*, 1984.

[24] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, Chapter 5, Kluwer Academic Publishers, 1996.

[25] J. Jubin and J.D. Tornow, "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, Jan. 1987.

[26] Z. Li and J. J. Garcia-Luna-Aceves, "A Distributed Approach For Multi-Constrained Path Selection And Routing Optimization," *Proc. IEEE QSHINE'06*, Aug. 2006.

[27] G. Malkin, "RIP Version 2" *RFC 2453*, 1998.

[28] M. Marina and S. Das, "Ad Hoc On-Demand Multipath Distance vector Routing," *Wireless Commun. Mob. Comput.*, 2006.

[29] J. M. McQuillan, I. Richer, and E. C. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Trans. Communications*, 1980.

[30] J. Moy, "OSPF Version 2" *RFC 2328*, 1998.

[31] M. Mosko and J.J. Garcia-Luna-Aceves, "Multipath Routing in Wireless Mesh Networks," *IEEE WiMesh '05*, Sept. 2005.

[32] S. Nelakuditi et al., "Adaptive Proportional Routing: A Localized QoS Routing Approach," *IEEE/ACM Trans. Networks*, 10(6):790–804, 2002.

[33] C. E. Perkins and P. Bhagwat, "Routing over Multihop Wireless Network of Mobile Computers," *Proc. ACM SIGCOMM '94*, 1994.

[34] C. E. Perkins and E. M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. IEEE WMCSA '99*, 1999.

[35] D. Savage et al., "Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)" *RFC 7868*, 2016.

[36] M. Schwartz and T. Stern, "Routing Techniques Used in Computer Communication Networks," *IEEE Trans. Communications*, Apr. 1980.

[37] A. Segall, "Distributed Network Protocols," *IEEE Trans. on Information Theory*, Jan. 1983.

[38] B.R. Smith and J.J. Garcia-Luna-Aceves, "Best Effort Quality-of-Service," *Proc. IEEE ICCCN '08*, Aug. 2008.

[39] B.R. Smith and J. Samson, "Herding packets: Properties Needed of Metrics for Loop-Free & Best Forwarding Paths," *Proc. IEEE ICNC '17*, 2017.

[40] J. L. Sobrinho, "An Algebraic Theory of Dynamic Network Routing," *IEEE/ACM Trans. Networking*, Oct. 2005.

[41] J. L. Sobrinho and M.A. Ferreira, "Routing on Multiple Optimality Criteria," *Proc. ACM SIGCOMM '20*, August 10–14, 2020.

[42] M. Spohn and J.J. Garcia-Luna-Aceves, "Neighborhood Aware Source Routing," *Proc. ACM MobiHoc '01*, Oct. 4–5, 2001.

[43] R. Van Glabbeek et al., "Sequence Numbers Do Not Guarantee Loop Freedom–AODV Can Yield Routing Loops," *Proc. ACM MSWiM '13*, Nov. 2013.

[44] P. Van Mieghem, H. D. Neve, and F. Kuipers, "Hop-by-Hop Quality of Service Routing," *Computer Networks*, 2001.

[45] S. Vutukury and J.J. Garcia-Luna-Aceves, "A Simple Approximation to Minimum-Delay Routing," *Proc. ACM SIGCOMM '99*, Aug. 31–Sept. 3, 1999.

[46] S. Vutukury and J.J. Garcia-Luna-Aceves, "MDVA: A Distance-Vector Multipath Routing Protocol," *Proc. IEEE INFOCOM '01*, April 2001.

[47] J. Wang and K. Nahrstedt, "Hop-by-Hop Routing Algorithms for Premium-class Traffic in Diffserv Networks," *Proc. IEEE INFOCOM '02*, 2002.

[48] Y. Wang, M. Schapira, and J. Rexford, "Neighbor-specific BGP: More Flexible Routing Policies While Improving Global Stability," *Proc. ACM SIGMETRICS '09*, 2009.

[49] Z. Wang and J. Crowcroft, "Quality-of-Sservice Routing for Supporting Multimedia Applications," *IEEE Journal of Selected Areas in Communications*, 1996.

[50] X. Wu, et al., "A Unified Analysis of Routing Protocols in MANETs," *IEEE Trans. Commun.*, March 2010.

[51] W. Xu and J. Rexford, "MIRO: Multi-path Interdomain Routing," *Proc. ACM SIGCOMM '06*, 2006.

[52] X. Yuan, "Heuristic Algorithms for Multiconstrained Quality-of-Service Routing," *IEEE/CM Trans. on Networking*, April 2002.

[53] W.T. Zaumen and J.J. Garcia-Luna-Aceves, "Loop-Free Multipath Routing Using Generalized Diffusing Computations," *Proc. IEEE INFOCOM '98*, March 29–April 2, 1998.

[54] W.T. Zaumen and J.J. Garcia-Luna-Aceves, "System for Maintaining Multiple Loop-Free Paths between Source Node and Destination Node in Computer Network," U.S. Patent 5,881,243, March 9, 1999.