Wheel + Ring = Reel: the Impact of Route Filtering on the Stability of Policy Routing

Luca Cittadini Giuseppe Di Battista Massimo Rimondini Stefano Vissicchio
Dept. of Computer Science and Automation, Roma Tre University
{ratm,gdb,rimondin,vissicch}@dia.uniroma3.it

***Abstract*—BGP allows providers to express complex routing policies preserving high degrees of autonomy. However, unrestricted routing policies can adversely impact routing stability. A key concept to understand the interplay between autonomy and expressiveness on one side, and stability on the other side, is safety under filtering, i.e., guaranteed stability under autonomous usage of route filters. BGP route filters are used to selectively advertise specific routes to specific neighbors.**

We provide a necessary and sufficient condition for safety under filtering, filling the large gap between previously known necessary and sufficient conditions. Our characterization is based on the absence of a particular kind of dispute wheel, a structure involving circular dependencies among routing preferences. We exploit our result to show that networks admitting multiple stable states are provably unsafe under filtering. This is especially interesting from an operational point of view, since networks with multiple stable states actually happen in practice (BGP wedgies). Finally, we show that adding filters to an existing configuration may lead to oscillations even if the configuration is safe under any link failure. Unexpectedly, we find policy configurations where misconfigured filters can do more harm than network faults.

I. Introduction and Related Work

Network operators can control interdomain routing at a fine grained level by relying on the policy-based Border Gateway Protocol (BGP) [1]. BGP provides Autonomous Systems (ASes) with the *autonomy* to set routing policies independent of each other, and with the *expressiveness* to specify extremely complex configurations. However, autonomy and expressiveness come at the expense of guaranteed convergence. In particular, a BGP configuration could never reach a stable routing, either because a stable state for that configuration does not exist at all, or because the protocol gets "trapped" into bad event timings. This is highly undesirable, since it has been observed that interdomain routing changes can cause performance degradation and packet loss [2], and continuously changing routing can severely affect the availability of services [3]. The need to avoid such disadvantages has spurred significant research efforts on BGP stability.

Varadhan et al. [4] showed that autonomy in configuring routing policies can lead to persistent routing oscillations, and proposed constraints to be applied to routing policies in order to achieve *safety*, i.e., stability under any timings of routing events. A number of fundamental contributions on this topic are due to Griffin et al. [5], [6], [7], [8]. Among the results they presented, those works showed how the dynamic behavior of BGP can be related to characteristics of the BGP configuration that can be statically analyzed. In particular, in [8] it is shown that the absence of a *dispute wheel* (DW), a cyclic pattern of routing preferences, is sufficient to guarantee safety.

The "no DW" condition is a cornerstone in the literature on BGP stability. As an example, Gao et al. [9], [10] used the absence of DWs to prove that, if routing policies are specified consistently with the commercial relationships between ASes, safety is guaranteed.

The stability of path vector protocols has also been studied using algebraic approaches in [11], [12], [13], [14]. These works described convergence conditions that are based on properties of path rankings, and showed that the no DW condition finds a counterpart also in algebraic models. In particular, in [13] the authors proposed a relaxation of the guidelines presented in [9].

In [15] Chau took into account the general case in which non-strict path rankings can be expressed. Even in this case, the absence of DWs is fundamental to guarantee safety.

Feamster et al. [16], [17] explored the impact of autonomy and expressiveness on the stability of the BGP protocol. In these papers the roles of the *ranking* and *filtering* components of routing policies are clearly distinguished. Ranking allows an AS to specify preferences over multiple candidate routes to the same destination, while filtering allows an AS to selectively advertise specific routes to specific neighbors. A crucial question is posed in [17]: "provided that each AS retains complete autonomy and complete filtering expressiveness, how expressive can rankings be while guaranteeing stable routing?". This question is formalized by the concept of *safety under filtering*. A configuration is safe under filtering if it is safe under any combination of route filters. A necessary condition for safety under filtering is the absence of a particular subclass of DWs, called *dispute rings* [17].

In this paper, we make three main contributions. First, we show a necessary and sufficient condition for safety under filtering, filling the large gap between previously known necessary and sufficient conditions. To the best of our knowledge, this is the first characterization of stability in policy routing under realistic assumptions about the autonomy of ASes. Our result is based on the presence of a structure called *dispute reel* (DR), which is both a special case of a DW *and* a generalization of a dispute ring. Dispute reels inherit from DWs the interesting property of depending on routing policies alone. Hence, checking for the presence of a DR does not require to delve into the details of BGP dynamics.

Second, we show that, in a network admitting multiple stable routing states, safety under filtering is provably compromised. In particular, whenever the existence of multiple stable states is detected, we provide a way for network operators to pinpoint the portions of the BGP configuration which define a DR (thus making the configuration not safe under filtering). Observe that this implies that the so called BGP wedgies [18] are an hallmark for unsafety under filtering.

Third, we show that robustness does not necessarily imply safety under filtering. *Robustness* is the property of a configuration to be safe under any combination of link/node failures [8]. It is known that safety under filtering implies robustness [17]. We explore the relationship between those two properties by showing that the opposite does not hold. In a sense, this proves that the autonomy of adding (possibly misconfigured) filters can do more harm than network faults.

The popularity of DWs in the literature on the stability of policy-based protocols is mostly due to the fact that the "no DW" condition implies the existence of a unique stable routing state [8], safety [8], robustness [8], and safety under filtering [17]. As a side effect of our work, we show that DRs can replace DWs, giving raise to less constraining sufficient conditions for all those properties.

The paper is organized as follows. Sec. II introduces SPVP, which is a commonly adopted model to study BGP stability. Sec. III defines the concept of dispute reel. Sections IV and V respectively present the necessary and sufficient conditions of our characterization of safety under filtering. In Sec. VI we discuss the relationship between safety under filtering and robustness. Conclusions are drawn in Sec. VII.

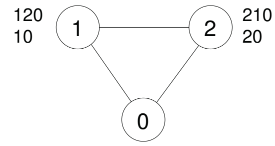
II. A Model for Policy-Based Path Vector Routing Protocols

In this section we describe SPVP (*Simple Path Vector Protocol*), a model that is widely used [7], [6], [10], [8] to study BGP stability and that we adopt throughout the paper.

In SPVP the topology of an interdomain routing system is modeled as a graph. Let $G = (V, E)$ be a simple undirected graph with vertex set $V = \{0, 1, \ldots, n\}$ and edge set E. A *path* P in G is either the empty path, denoted by ϵ, or a sequence of $k+1$ vertices $P = (v_k\ v_{k-1}\ \ldots\ v_0)$, $v_i \in V$ such that $(v_i, v_{i-1}) \in E$ for $0 < i \leq k$. When it is clear from the context, we will treat sequences as sets, e.g., we will use $v \in P$ to denote that v appears in path P. Vertex v_{k-1} is the *next hop* of v_k in P. If $k = 0$ then (v_0) is a path consisting of a single vertex v_0. The *concatenation* of two nonempty paths $P = (v_k\ v_{k-1}\ \ldots\ v_i)$, $k \geq i$, and $Q = (v_i\ v_{i-1}\ \ldots\ v_0)$, $i \geq 0$, denoted as PQ, is the path $(v_k\ v_{k-1}\ \ldots\ v_i\ v_{i-1}\ \ldots\ v_0)$. Since the empty path represents unreachability, it can never be extended, that is, $P\epsilon = \epsilon P = \epsilon$.

In SPVP every vertex attempts to establish a path to vertex 0 relying on the paths used by its neighbors. Each vertex $v \in V$ is assigned a set of *permitted paths* $\mathcal{P}^v$. All these paths start from v and end in 0 and represent the paths that v can use to reach 0. Let $\mathcal{P}^0 = \{(0)\}$ and let $\mathcal{P} = \bigcup_{v \in V} \mathcal{P}^v$.

For each vertex $v \in V$, a *ranking function* $\lambda^v : \mathcal{P}^v \to \mathbb{N}$ determines the relative level of preference $\lambda^v(P)$ assigned by



	0	1	2
π_1	(0)	(1 0)	(2 1 0)
π_2	(0)	(1 2 0)	(2 0)

Fig. 1. Instance DISAGREE (on the left) admits two distinct stable states π_1 and π_2 (on the right). Each row of the table represents a state. Each column specifies the path selected by every vertex in this state.

v to path P. If $P_1, P_2 \in \mathcal{P}^v$ and $\lambda^v(P_2) < \lambda^v(P_1)$, then P_2 is *preferred* over P_1. Let $\Lambda = \{\lambda^v | v \in V\}$.

For each $v \in V - \{0\}$, the following conditions are assumed to hold on the paths:

i) $\epsilon \in \mathcal{P}^v$;
ii) $\forall P \in \mathcal{P}^v$ with $P \neq \epsilon$: $\lambda^v(\epsilon) > \lambda^v(P)$;
iii) $\forall P_1, P_2 \in \mathcal{P}^v, P_1 \neq P_2 : \lambda^v(P_1) = \lambda^v(P_2) \Rightarrow P_1 = (v\ u)P_1', P_2 = (v\ u)P_2'$, i.e., v has the same next hop in P_1 and in P_2; and
iv) $\forall P \in \mathcal{P}^v$: P is simple (i.e., has no repeated vertices).

Since the empty path represents unreachability of 0, Condition *i* states that all vertices but 0 may be unable to reach the destination. Unreachability is the last chance for a vertex (Condition *ii*). Condition *iii* states that function λ^v induces a total order on all the paths of $\mathcal{P}^v$ but those that begin with the same pair of vertices. Such paths reach 0 via the same neighbor of v, and can therefore be considered equivalent. Condition *iv* accounts for acyclic routing.

An instance of SPVP is a triple $S = (G, \mathcal{P}, \Lambda)$, where $G = (V, E)$ is a simple undirected graph, $\mathcal{P}$ is the set of permitted paths, and Λ is the set of ranking functions. Fig. 1 shows an instance of SPVP, called DISAGREE [8]. The graphical convention we use in this figure will be adopted throughout the paper. Namely, each vertex v is equipped with a list of paths representing $\mathcal{P}^v$ sorted by increasing values of λ^v. The empty path and $\mathcal{P}^0$ are omitted for brevity.

An instance of SPVP provides a good abstraction of an interdomain routing system. In fact, vertices can be mapped to Autonomous Systems, edges can be interpreted as peering sessions, and permitted paths, together with the rankings, can represent routing policies.

The dynamic behavior of BGP is modeled by the distributed asynchronous algorithm [7], [6] shown in Fig. 2. In the following we will refer both to the model introduced so far and to this algorithm using the name SPVP. In order to establish a path to 0, in the SPVP algorithm vertices exchange messages containing permitted paths. It is assumed that message exchanges are reliable and edges introduce an arbitrary finite delay. Communication between vertices takes place in a totally asynchronous way.

To describe the SPVP algorithm in Fig. 2 we need a few more definitions. Let peers(v) be the set of neighbors of v. Two data structures are used at each vertex v to represent the information v is aware of at time t: the path rib$_t(v)$ that is used to reach 0 and a table rib-in$_t(v \Leftarrow u)$ that stores the latest path received from neighbor $u \in$ peers(v). Thus, vertex

```
process spvp(v)
1: while receive P from u do
2:     rib-in_t(v ⇐ u) := P
3:     rib_t(v) := best_t(v)
4:     if rib_t(v) ≠ best_{t−1}(v) then
5:         for all v ∈ peers(v) do
6:             send rib_t(v) to v
7:         end for
8:     end if
9: end while
```

Fig. 2. A distributed asynchronous algorithm (SPVP) for modeling the dynamic behavior of BGP.

v can select a path to 0 among the choices available in

$$\text{choices}_t(v) = \{(v\ u)P \in \mathcal{P}^v \mid P = \text{rib-in}_t(v \Leftarrow u)\}$$

Let W be a subset of the permitted paths $\mathcal{P}^v$ at vertex v, such that each path in W has a distinct next hop. Then the *best* path at v in W is

$$\text{best}(W, v) = \begin{cases} P \in W | P = \arg\min\ \lambda^v(P) & (W \neq \oslash) \\ \epsilon & (W = \oslash) \end{cases}$$

and the overall best path v is aware of at time t is $\text{best}_t(v) = \text{best}(\text{choices}_t(v), v)$.

In SPVP each vertex executes an instance of the algorithm in Fig. 2. When a vertex v receives a path P from one of its neighbors u, it stores P in the local data structure $\text{rib-in}_t(v \Leftarrow u)$ and recomputes its best path. If the computed best path Q differs from the previously selected path, u sends a message containing P to all of its neighbors.

We say that an instance S of SPVP is *consistent* if, for any $u \in V$ and $P \in \mathcal{P}^u$, we have that $P = (u\ v)P'$ and $P' \in \mathcal{P}^v$. We stress that the presence of a path violating this condition cannot affect the behavior of the SPVP algorithm on S. Therefore, without loss of generality we assume throughout the paper that all SPVP instances are consistent.

A *path assignment* is a function π that maps each vertex $v \in V$ to a path $\pi(v) \in \mathcal{P}^v$. We have that $\pi(0) = (0)$ and, if $\pi(v) = \epsilon$, then v cannot reach vertex 0. In the following, we will refer to π as the *state* of the SPVP instance. Observe that, at any time t, the algorithm in Fig. 2 defines a path assignment π_t where $\pi_t(v) = \text{rib}_t(v)$ and each vertex always selects the best available path.

We say that an *edge* (u, v) is *activated* at time t from u to v if v executes the algorithm in Fig. 2 to process the latest message received from u. The order in which protocol messages are exchanged needs not to be total, i.e., at a given instant more than one message can be processed. This order is represented using activation sequences. An *activation sequence* σ is a (possibly infinite) sequence of sets $\sigma = (A_0\ A_1\ \ldots\ A_i \ldots)$ in which each set A_t contains the edges that are activated at time t. Each edge in A_t is considered oriented according to the direction of the activation. Given an SPVP instance S, we say that an activation sequence σ on S *leads to* path assignment π_{t_2} *starting from* path assignment π_{t_1}, denoted by $\pi_{t_1} \overset{\sigma}{\rightsquigarrow} \pi_{t_2}$, if, after activating edges according to σ, S changes its state from π_{t_1} to π_{t_2}.

t	A_t	1	2
1	$\{(0,1),(0,2)\}$	*(1 0)*	*(2 0)*
2	$\{(1,2),(2,1)\}$	*(1 2 0)* (1 0)	*(2 1 0)* (2 0)
3	$\{(1,2),(2,1)\}$	*(1 0)*	*(2 0)*

TABLE I
AN INFINITE FAIR ACTIVATION SEQUENCE FOR DISAGREE (FIG. 1).

We say that an activation sequence is *fair* [8] if, whenever vertex u sends a message at time t (Step 6 of SPVP), there exists a time $t' > t$ at which the message is delivered and processed by its recipient. This means that edge (u, v) is eventually activated when u sends a message to v.

Observe that there exist variants of the SPVP algorithm (e.g., considered in [19], [7], [9], [17]) where only some classes of activation sequences are allowed. For the sake of completeness, we base our study on the original version of the algorithm, in which edges are activated independently and simultaneous activations are allowed. It has been shown [20] that any relaxed version of this model is only able to capture a strictly smaller set of routing oscillations.

A state $\pi_{t'}$ of an SPVP instance is a *stable state* if $\forall v \in V$: $\pi_t(v) = \pi_{t'}(v)$ for any $t > t'$. For example, two stable states for DISAGREE are described in Fig. 1.

There are SPVP instances that do not have any stable states (e.g., BAD-GADGET in [5]), and deciding whether an SPVP instance admits a stable path assignment is NP-complete [8].

Even if a stable state exists, routing protocols can still get trapped [5], [8] into persistent oscillations. An SPVP instance is *safe* if any fair activation sequence eventually leads to a stable state. Determining whether an SPVP instance is safe is referred to as the *safety* problem. For example, as detailed in Tab. I, DISAGREE is not safe, since there exists an infinite fair activation sequence never leading to a stable state. The activation sequence is represented in a tabular notation: each row corresponds to an activation, the first column represents time, the second one specifies activated edges, and the remaining columns represent the current rib-in_t at each vertex, with the currently selected best path highlighted using italic face. The initial state is assumed to be $\pi_0(v) = \epsilon\ \forall v \in V - \{0\}$.

We can now formally define the concept of safety under filtering, already discussed in the Introduction.

Definition 2.1: An SPVP instance $S = (G, \mathcal{P}, \Lambda)$ is *safe under filtering* (SUF) if, for any $\mathcal{P}' \subseteq \mathcal{P}$, the instance $(G, \mathcal{P}', \Lambda)$ is safe.

III. WHEEL + RING = REEL

It is shown in [17] that safety under filtering can be studied by analyzing structural properties of the policy configuration, without the need to deal with the details of dynamic evaluation. The main known structural properties that are related to safety under filtering are based on the absence of cyclic

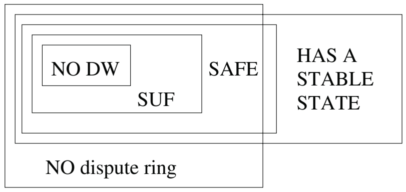


Fig. 3. The absence of a dispute ring (wheel) is a necessary (sufficient) condition for safety under filtering [17].

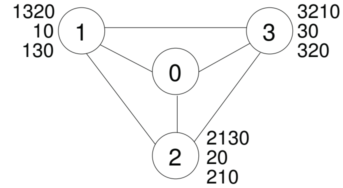


Fig. 4. An SPVP instance, showed in [17], which is safe under filtering but contains DWs. However, none of these DWs is a DR.

dependencies among routing preferences, which are called dispute wheels and dispute rings.

A *dispute wheel* (DW) [8] $\Pi = (\vec{\mathcal{U}}, \vec{\mathcal{Q}}, \vec{\mathcal{R}})$ is a triple consisting of a sequence of nodes $\vec{\mathcal{U}} = (u_0\ u_1 \ldots u_{k-1})$ and two sequences of nonempty paths $\vec{\mathcal{Q}} = (Q_0\ Q_1 \ldots Q_{k-1})$ and $\vec{\mathcal{R}} = (R_0\ R_1 \ldots R_{k-1})$ such that for each $i = 0, \ldots, k-1$ we have: *i*) R_i is a path from u_i to u_{i+1} *ii*) $Q_i \in \mathcal{P}^{u_i}$ *iii*) $R_i Q_{i+1} \in \mathcal{P}^{u_i}$ *iv*) $\lambda^{u_i}(R_i Q_{i+1}) \le \lambda^{u_i}(Q_i)$

We call vertices u_i *pivot* vertices, paths Q_i *spoke* paths, and paths R_i *rim* paths. Throughout the paper, we intend subscripts of vertices and paths in a dispute wheel to be interpreted modulo k where $k = |\vec{\mathcal{U}}|$. The absence of a dispute wheel implies safety under filtering [8], [17].

Feamster et al. show in [17] that the absence of a particular class of dispute wheels, called *dispute rings*, is a necessary condition for safety under filtering. A dispute ring is a dispute wheel having at least three pivot vertices, and such that each vertex appears only once in the wheel. Fig. 3 shows how the "no dispute wheel" and "no dispute ring" conditions relate to the properties of an SPVP instance. We stress that there is a large gap between the two conditions, as the absence of a dispute ring does not guarantee safety, and does not even imply that the SPVP instance admits a stable path assignment.

We now define a dispute reel as a special case of dispute wheel. Intuitively, a reel is a dispute wheel such that the spoke paths form a tree T and each rim path R_i contains no vertex in T except u_i and u_{i+1}. In order to formally define the dispute reel, we use the notation $P[v]$ to denote the subpath of P starting at vertex v, that is, $P = (u \ldots v)P[v]$. This implies $P[0] = (0)$ for any P.

Definition 3.1: A *dispute reel* (DR) is a dispute wheel which satisfies the following conditions:

i) (*Pivot vertices appear in exactly three paths*) – for each $u_i \in \vec{\mathcal{U}}$, u_i only appears in paths Q_i, R_i and R_{i-1}.

ii) (*Spoke and rim paths do not intersect*) – for each $u \notin \vec{\mathcal{U}}$, if $u \in Q_i$ for some i, then no j exists such that $u \in R_j$.

iii) (*Spoke paths form a tree*) – for each distinct $Q_i, Q_j \in \vec{\mathcal{Q}}$, if $v \in Q_i \cap Q_j$, then $Q_i[v] = Q_j[v]$.

We stress that the existence of a DR does not depend at all on the protocol dynamics, i.e., it is a structural property of the policy configuration that can be statically checked. It is easy to check that DISAGREE (Fig. 1) is an example of a DR. Conversely, the instance in Fig. 4, first used in [17] to show that the presence of a DW does not prevent an instance from being safe under filtering, does not contain any DRs. As an example, a DW Π exists in Fig. 4 where pivot vertices are $\vec{\mathcal{U}} = (1\ 2\ 3)$, spoke paths are $\vec{\mathcal{Q}} = ((1\ 0)\ (2\ 0)\ (3\ 0))$, and rim paths are $\vec{\mathcal{R}} = ((1\ 3\ 2)\ (2\ 1\ 3)\ (3\ 2\ 1))$. However, pivot vertex 1 appears in all rim paths, thus violating Condition *i* of Definition 3.1. On the other hand, the instance in Fig. 4 also contains the DW Π' where pivot vertices are $u_0 = 1$ and $u_1 = 2$, spoke paths are $Q_0 = (1\ 3\ 0)$ and $Q_1 = (2\ 0)$, and rim paths are $R_0 = (1\ 3\ 2)$ and $R_1 = (2\ 1)$. Π' too is not a DR because Condition *ii* is not satisfied, as vertex 3 appears both in Q_0 and in R_0. Similar arguments can be applied to the other DWs in the instance in Fig. 4.

An even simpler dispute wheel is the *dispute duo*.

Definition 3.2: A *dispute duo* is a dispute reel such that $|\vec{\mathcal{U}}| = 2$ and $R_0 \cap R_1 = \{u_0, u_1\}$.

The simple structure of DRs allows us to identify two classes of activation sequences leading to two "natural" classes of path assignments. Given an SPVP instance S containing a DW Π, the *supporting instance* $S[\Pi]$ *of* Π is the minimal SPVP instance which contains the vertices, edges and paths of Π. Intuitively, $S[\Pi]$ can be obtained from S by filtering all paths but those used in the DW. Observe that, if Π is a DR, then in $S[\Pi]$ pivot vertices have exactly two permitted paths, and vertices along the spoke paths (except pivots) have exactly one permitted path.

Let S be an SPVP instance containing a DR Π and let $S[\Pi]$ be the supporting instance of Π. The *all-spoke* path assignment (see Fig. 5(a)) is a path assignment $\bar{\pi}$ such that $\bar{\pi}(u) = Q_i[u]$ if $u \in Q_i$, $\bar{\pi}(u) = \epsilon$ otherwise. Since spoke paths form a tree, by activating the edges of each spoke path Q_i in reverse order (starting from 0) it is easy to construct an activation sequence σ_{spoke} leading to an all-spoke path assignment.

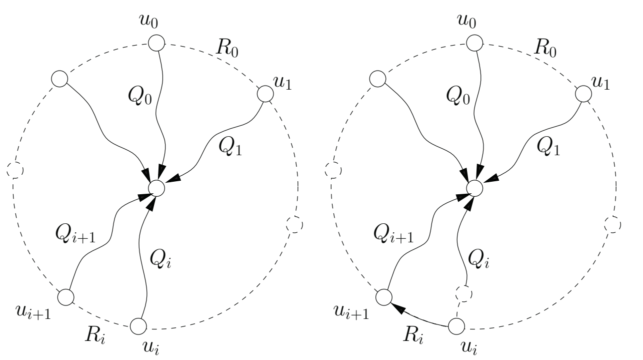
Similarly, we define the *one-rim* path assignment *for pivot* u_i (see Fig. 5(b)) as a path assignment $\bar{\pi}^i$ such that:

$$\bar{\pi}^i(u) = \begin{cases} Q_j[u] & \text{if } u \in Q_j, u \neq u_i \\ R_i[u]Q_{i+1} & \text{if } u \in R_i \\ \epsilon & \text{otherwise.} \end{cases}$$

In order to build an activation sequence that leads to $\bar{\pi}^i$, we can extend σ_{spoke} by activating the edges of R_i in reverse order (starting from u_{i+1}). This is always possible because rim paths never intersect spoke paths and for each non-pivot vertex along R_i, $\bar{\pi}(v) = \epsilon$.

IV. Safety Under Filtering implies no DR

In this section we show that the absence of DRs is a necessary condition for safety under filtering. We do this by



(a) All-spoke path assignment $\bar{\pi}$. (b) One-rim path assignment $\bar{\pi}^i$.

Fig. 5. Two special path assignments of a dispute reel. The selected paths are highlighted using solid stroke. Note that in $\bar{\pi}^i$, u_i is the only vertex in Q_i which is not selecting a subpath of Q_i.

showing that the presence of a DR in an SPVP instance S makes S not safe under filtering. The proof consists of three parts. First, we show that if S contains a dispute duo, then S is not SUF (Lemma 4.1). Second, we generalize this result by stating that if S contains a DR consisting of two pivot vertices, then S is not SUF (Lemma 4.2). Last, we show that if an instance S contains a DR Π, then an oscillation can always be constructed, either by cycling through one-rim path assignments on Π (Lemma 4.3), or by exploiting a different DR consisting of two pivot vertices (Lemma 4.4). Thus, S is not safe under filtering.

A. Dispute Reels with 2 Pivots

We start by showing that the presence of a dispute reel having 2 pivot vertices makes an SPVP instance not safe under filtering. First, we generalize the routing oscillation showed in Tab. I for DISAGREE to the broader class of dispute duos.

Lemma 4.1: An SPVP instance that contains a dispute duo is not safe under filtering.

Proof: Let S be an SPVP instance containing a dispute duo $\Pi = (\vec{\mathcal{U}}, \vec{\mathcal{Q}}, \vec{\mathcal{R}})$ and consider $S[\Pi]$. We now construct a fair activation sequence that induces an oscillation on $S[\Pi]$. The main idea is that vertices u_0 and u_1 can simultaneously select paths $\pi(u_0) = R_0Q_1$ and $\pi(u_1) = R_1Q_0$. Path assignment π is clearly not stable, so the two pivot vertices will eventually fall back on their spoke paths Q_0 and Q_1. By iterating this argument, we are able to show an infinite fair activation sequence.

First of all, since Π is a DR, we can construct on $S[\Pi]$ an activation sequence that leads to the all-spoke path assignment π_{t_1} at some time t_1. We now propagate the announcement of path Q_1 (respectively, Q_0) by activating the edges along R_0 (R_1) in reverse order. Since R_0 and R_1 have no shared vertices other than u_0 and u_1, the two announcements cannot interfere with each other. We halt one hop before the announcement of Q_1 (Q_0) reaches u_0 (u_1). Formally, let $R_0 = (v_0\ v_1 \dots v_k)$, where $v_0 = u_0$ and $v_k = u_1$. We activate edges in R_0 in reverse order until we hit v_1, that is,

$$\sigma_{R_0} = (\{(v_k, v_{k-1})\}\ \{(v_{k-1}, v_{k-2})\} \dots \{(v_2, v_1)\}).$$

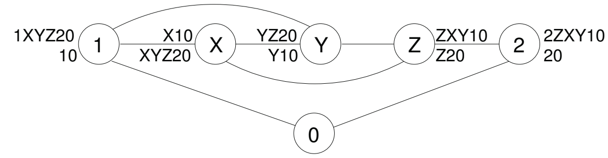


Fig. 6. An SPVP instance containing a DR consisting of two pivot vertices (1 and 2) and whose rim paths intersect at vertices X, Y, and Z.

Symmetrically, let $R_1 = (w_0\ w_1 \dots w_j)$, and consider the sequence

$$\sigma_{R_1} = (\{(w_j, w_{j-1})\}\ \{(w_{j-1}, w_{j-2})\} \dots \{(w_2, w_1)\}).$$

We activate edges according to σ_{R_0}, and then according to σ_{R_1}. Then, we simultaneously activate edges (v_1, v_0) and (w_1, w_0). Observe that the simultaneous activation of edges (v_1, v_0) and (w_1, w_0) makes path R_0Q_1 available at u_0, and path R_1Q_0 available at u_1. It is easy to check that these activations lead to a path assignment π_{t_2} such that, for $i \in \{0, 1\}$:

$$\pi_{t_2}(u) = \begin{cases} Q_i[u] & \text{if } u \in Q_i, u \neq u_i \\ R_i[u]Q_{i+1} & \text{if } u \in R_i \end{cases}$$

We now activate edges in R_0 (R_1) in reverse order, again halting at v_1 (w_1), and then we simultaneously activate edges (v_1, v_0) and (w_1, w_0). By doing so, vertex u_0 (u_1) withdraws the availability of path Q_0 (Q_1). Since R_0 and R_1 do not have vertices in common other than u_0 and u_1, the withdrawal will eventually reach vertex u_1 (u_0). Vertex u_1 (u_0) will then fall back on path Q_1 (Q_0). Observe that we have now reached the all-spoke path assignment π_{t_3}, which implies $\pi_{t_3}(u) = \pi_{t_1}(u)$ for every vertex u. Since we can iterate this argument, it is clear that there exists an infinite activation sequence. Moreover, no announcement is delayed indefinitely, i.e., the activation sequence is also fair on $S[\Pi]$. The proof is completed by noting that $S[\Pi]$ can be obtained by S through path filtering, hence we conclude that S is not SUF. ∎

Lemma 4.1 can be generalized, as DRs having two pivot vertices always imply the existence of a dispute duo. As an example, consider the instance in Fig. 6. Clearly, this instance contains a DR having $u_0 = 1$ and $u_1 = 2$ as pivot vertices, $Q_0 = (1\ 0)$ and $Q_1 = (2\ 0)$ as spoke paths, and $R_0 = (1\ X\ Y\ Z\ 2)$ and $R_1 = (2\ Z\ X\ Y\ 1)$ as rim paths. Notice that both rim paths traverse vertices X, Y, and Z. We now search for a dispute duo. Walk along R_1 and stop at the last vertex which is in $R_1 \cap R_0$, that is, Y. By analyzing λ^Y, it is easy to see that there exists another DR having Y and 2 as pivot vertices, $(Y\ 1\ 0)$ and $(2\ 0)$ as spoke paths, and $(Y\ Z\ 2)$ and $(2\ Z\ X\ Y)$ as rim paths. Note that the rim paths of this DR do not intersect at vertex X. We now repeat the process on the new DR, considering vertex Z. It is easy to see that there exists a dispute duo having Z and Y as pivot vertices. The following lemma generalizes the approach we just showed to any DR having two pivot vertices.

Lemma 4.2: An SPVP instance that contains a dispute reel having exactly 2 pivot vertices is not safe under filtering.

Proof: Let S be an SPVP instance containing a dispute reel $\Pi = (\vec{\mathcal{U}}, \vec{\mathcal{Q}}, \vec{\mathcal{R}})$, with $|\vec{\mathcal{U}}| = 2$. First, we show that the presence of Π implies that S contains a dispute duo Π', then we use Lemma 4.1 to argue that S is not SUF.

If R_0 and R_1 do not share any vertices except u_0 and u_1, then Π is a dispute duo and the statement directly follows from Lemma 4.1. Otherwise, let $\{v_1, \dots v_k\}$ be the vertices in $R_0 \cap R_1 - \{u_0, u_1\}$, in the same order as they appear in R_0. That is, $R_0 = (u_0 \dots v_1 \dots v_k \dots u_1)$, where $\forall i\ v_i \in R_1$. Let v_j be the "rightmost" vertex in R_1 among vertices $\{v_1, \dots, v_k\}$, and let $P = R_1[v_j]$. More formally, v_j is such that $v_i \notin P\ \forall i \neq j$. We now show that either there exists a dispute duo Π' having u_0 and v_j as pivot vertices, or there exists a DR Π'' consisting of two pivot vertices v_j and u_1 and having strictly less intersections between its rim paths than Π.

Refer to Fig. 7. Split R_1 and R_2 such that $R_1 = A(v_j)P$ and $R_0 = R(v_j)Q$.

Since we are considering $S[\Pi]$ and $v_j \in R_0 \cap R_1$, $\mathcal{P}^{v_j} = \{PQ_0, QQ_1\}$. Depending on the ranking at vertex v_j and since (by construction) we cannot have $\lambda^{v_j}(PQ_0) = \lambda^{v_j}(QQ_1)$, we have two possible cases.

i) $\lambda^{v_j}(PQ_0) < \lambda^{v_j}(QQ_1)$. We now show that $\Pi' = ((u_0\ v_j), (Q_0\ QQ_1), (R\ P))$ is a dispute duo. By construction, Π' has only two pivot vertices, and $P \cap R = \{u_0, v_j\}$. Observe that u_0 appears only in Q_0, R and P, while v_j appears only in QQ_1, R, and P. Therefore, Condition *i* of Definition 3.1 is satisfied. Condition *ii* is also satisfied, since $Q_0 \cap R = Q_0 \cap P = \{u_0\}$ and $Q_1 \cap R = Q_1 \cap P = \oslash$ are guaranteed by the fact that Π is a DR. Moreover, by construction, $Q \cap R = Q \cap P = \{v_j\}$. Finally, Condition *iii* holds for paths Q_0 and Q_1 since Π is a DR, and $Q \cap Q_0 = \oslash$.

ii) $\lambda^{v_j}(PQ_0) > \lambda^{v_j}(QQ_1)$. We now show that $\Pi'' = ((v_j\ u_1), (PQ_0\ Q_1), (Q\ A))$ is a dispute reel. Since $v_j \neq u_0$ by construction, Π'' has strictly less intersections between rim paths than Π. Observe that v_j appears only in PQ_0, Q, and A, while u_1 appears only in Q_1, Q, and A. Hence, Condition *i* of Definition 3.1 is satisfied. Condition *ii* is also satisfied, since $Q_0 \cap Q = Q_0 \cap A = \oslash$ and $Q_1 \cap Q = Q_1 \cap A = \{u_1\}$ are guaranteed by the fact that Π is a dispute reel. By construction, $P \cap Q = P \cap A = \{v_j\}$. Finally, Condition *iii* holds for paths Q_0 and Q_1 since Π is a DR, and $P \cap Q_1 = \oslash$.

Hence, in the first case we find a dispute duo Π'. In the second case, we find another dispute reel Π'' having two pivot vertices and having strictly less intersections between rim paths than Π. By iterating this argument, we eventually end up finding a dispute duo. We then use the result from Lemma 4.1 to prove that an instance containing a DR with two pivot vertices is not safe under filtering. ∎

B. Dispute Reels with more than 2 Pivots

The next step is to show that the presence of a dispute reel having more than two pivot vertices makes an SPVP instance not safe under filtering. We prove that in two parts. First, we introduce the concept of a "rim-by-rim" dispute reel, that is, a DR for which it is easy to construct a routing oscillation. Second, we show that the presence of a dispute reel which is not rim-by-rim implies the existence of a dispute reel having only two pivot vertices.

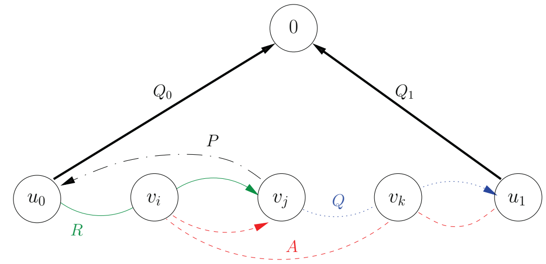


Fig. 7. A dispute reel having 2 pivot vertices. Rim paths $R_0 = RQ$ and $R_1 = AP$ are split as explained in the proof of Lemma 4.2. Different paths are represented using different strokes. In particular, spoke paths Q_0 and Q_1 are in thicker stroke.

Given a DR $\Pi = (\vec{\mathcal{U}}, \vec{\mathcal{Q}}, \vec{\mathcal{R}})$, with $|\vec{\mathcal{U}}| = k > 2$, we say that Π is *rim-by-rim* if $\forall i \in \{0, \dots, k-1\}$ there exists an activation sequence σ_i on $S[\Pi]$ such that $\bar{\pi}^i \overset{\sigma_i}{\rightsquigarrow} \bar{\pi}^{i+1}$. That is, starting from the one-rim path assignment for any pivot u_i, σ_i leads to the one-rim path assignment for pivot u_{i+1}. The following property is a straightforward consequence of the definition of rim-by-rim DR.

Property 4.1: σ_i activates all the edges in R_{i+1} at least once.

Observe that the well known instance BAD-GADGET, defined in [8], is a trivial rim-by-rim DR. More generally, any dispute ring can be viewed as a special case of rim-by-rim DR. Feamster et al. show in [17] that it is particularly easy to find an oscillation on a dispute ring. We are now able to generalize that result to the broader class of rim-by-rim DRs.

Lemma 4.3: An SPVP instance containing a rim-by-rim dispute reel is not safe under filtering.

Proof: Let S be an SPVP instance containing a rim-by-rim dispute reel Π. Using the fact that Π is rim-by-rim, we build an infinite fair activation sequence in the supporting instance $S[\Pi]$ that cycles indefinitely among one-rim path assignments.

As we have already seen, since Π is a dispute reel there exists an activation sequence on $S[\Pi]$ that induces a one-rim path assignment $\bar{\pi}^i$ for an arbitrary pivot u_i.

Since Π is rim-by-rim, there exist activation sequences σ_j such that $\bar{\pi}^i \overset{\sigma_i}{\rightsquigarrow} \bar{\pi}^{i+1} \overset{\sigma_{i+1}}{\rightsquigarrow} \dots \overset{\sigma_{i-1}}{\rightsquigarrow} \bar{\pi}^i$. Note that the initial and final path assignments are the same, thus we can iterate the same set of activations in order to create an infinite activation sequence σ on $S[\Pi]$. By Property 4.1, edges traversed by rim paths are activated at least once per iteration. To ensure fairness, at the end of each iteration we activate edges according to σ_{spoke} without altering the current path assignment. This implies that there exists an infinite fair activation sequence on $S[\Pi]$, hence S is not safe under filtering. ∎

Now consider the instance in Fig. 8. Clearly, this instance contains a DR Π where pivot vertices are $u_0 = 1$, $u_1 = 2$,

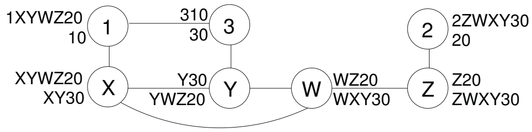


Fig. 8. A DR which is not rim-by-rim. Vertex 0 is omitted for brevity.

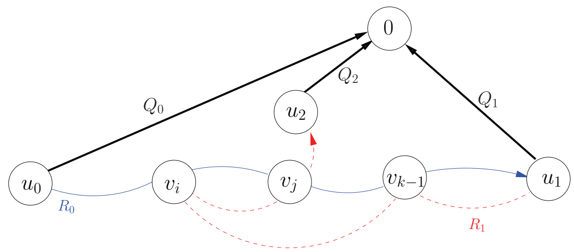


Fig. 9. A portion of a dispute reel which is not rim-by-rim, used in the proof of Lemma 4.4. Different paths are represented using different strokes. Spoke paths Q_0, Q_1, and Q_2 are in thicker stroke.

and $u_2 = 3$; spoke paths are $Q_0 = (1\ 0)$, $Q_1 = (2\ 0)$, and $Q_2 = (3\ 0)$; and rim paths are $R_0 = (1\ X\ Y\ W\ Z\ 2)$, $R_1 = (2\ Z\ W\ X\ Y\ 3)$, and $R_2 = (3\ 1)$. Π is not rim-by-rim: in particular, no activation sequence exists that, starting from the one-rim path assignment for pivot u_0 ($\bar{\pi}^0$), makes path R_1Q_2 available at vertex 2. In fact, assume that the instance is in state $\bar{\pi}^0$, that is, vertices 2 and 3 select their spoke paths, while vertices on R_0 select a subpath of R_0Q_1. In particular, vertex 1 selects path $(1\ X\ Y\ W\ Z\ 2\ 0)$. We now explore how far the announcement of path $(3\ 0)$ can be propagated along rim path R_1. Suppose that vertex 3 announces path $(3\ 0)$ to Y. Since path $(Y\ 3\ 0)$ is preferred, Y selects the new path and propagates the announcement to X. Observe that, even if X does not prefer path $(X\ Y\ 3\ 0)$, Y's announcement withdraws the availability of the previously selected path $(X\ Y\ W\ Z\ 2\ 0)$. Hence, X propagates the announcement further to W. Now, W does not change its choice, since path $(W\ X\ Y\ 3\ 0)$ is less preferred. It is easy to see that there is no way to propagate the announcement further than vertex W. Nevertheless, the rankings at vertex W are such that there exists a DR having W and 2 as pivot vertices. The following lemma shows that the presence of a DR having two pivot vertices is actually a general property of any DR which is not rim-by-rim. By using Lemma 4.2, we are then able to show an oscillation even on DRs that are not rim-by-rim.

Lemma 4.4: An SPVP instance containing a dispute reel which is not rim-by-rim is not safe under filtering.

Proof: Let S be an SPVP instance containing a dispute reel $\Pi = (\vec{\mathcal{U}}, \vec{\mathcal{Q}}, \vec{\mathcal{R}})$ which is not rim-by-rim. If $|\vec{\mathcal{U}}| = 2$, the statement follows from Lemma 4.2. Otherwise, consider $S[\Pi]$. Since Π is not rim-by-rim by hypothesis, there are at least $\bar{\pi}^i$ and $\bar{\pi}^{i+1}$ such that $\nexists \sigma : \bar{\pi}^i \overset{\sigma}{\rightsquigarrow} \bar{\pi}^{i+1}$. Assume, without loss of generality, that $i = 0$.

Let $\{v_1, \ldots v_k\}$ be the vertices of $R_0 \cap R_1$, in the same order as they appear in R_0, that is, $R_0 = (u_0 \ldots v_1 \ldots v_k)$, where $v_k = u_1$, as showed in Fig. 9.

Let Σ be the set of all the activation sequences that, starting from the one-rim path assignment $\bar{\pi}^0$, make path Q_2 available in the set of choices of some vertex v_m. More formally, $\forall \sigma \in \Sigma$, $\bar{\pi}^0 \overset{\sigma}{\rightsquigarrow} \pi_t$, where $R_1[v_m]Q_2 \in \text{choices}_t(v_m)$ for some m and t. Note that Σ contains at least the activation sequence obtained by activating the edges of R_1 in reverse order, which would lead to $R_1[v_j]Q_2 \in \text{choices}_t(v_j)$, where v_j is the common vertex that is the "rightmost" in R_1, that is, $\forall i \neq j$, $v_i \notin R_1[v_j]$. Consider the activation sequence $\sigma' \in \Sigma$ such that v_m has the highest index. We now show that, if the announcement of path Q_2 reaches vertex u_1, then we have a contradiction. In fact, if $v_m = u_1$, we would have $\bar{\pi}^0 \overset{\sigma'}{\rightsquigarrow} \pi_t$, where $\pi_t(u_1) = R_1Q_2$. This enable us to activate the edges in R_0 in reverse order, withdrawing the availability of path Q_1 on all the vertices along R_0, and eventually reaching state $\bar{\pi}^1$. This contradicts the hypothesis that $\nexists \sigma : \bar{\pi}^0 \overset{\sigma}{\rightsquigarrow} \bar{\pi}^1$.

Hence, $v_m \neq u_1$. We now prove that, if the announcement of path Q_2 cannot be propagated further than v_m, then we have a dispute reel having two pivot vertices. Consider the path ranking at vertex v_m. We have two possible cases:

i) $\lambda^{v_m}(R_1[v_m]Q_2) \leq \lambda^{v_m}(R_0[v_m]Q_1)$. We now show that there exists an activation sequence $\bar{\sigma} \in \Sigma$ that makes path Q_2 available in the set of choices of $v_{m'}$, with $m' > m$, hence a contradiction. Intuitively, v_m can announce path $R_1[v_m]Q_2$ to withdraw the availability of path $R_0[v_m]Q_1$ to the vertices on R_0. This allows the announcement of path Q_2 to be propagated beyond vertex v_m. Observe that, since path $R_1[v_m]Q_2$ is in $\text{choices}_t(v_m)$ and it is preferred, we must have $\pi_t(v_m) = R_1[v_m]Q_2$ after activation sequence σ'. Let σ_1 consist of the activations of all the edges in R_0 in reverse order, starting from v_m. Let π_{t_1} be the path assignment after σ_1, that is, $\pi_t \overset{\sigma_1}{\rightsquigarrow} \pi_{t_1}$. Note that π_{t_1} is such that path $R_0[v_h]Q_1$ has been withdrawn at each v_h, $h < m$. We now construct σ_2 by activating the edges along R_1 in reverse order. In this way, v_m propagates the announcement of path $R_1[v_m]Q_2$. Clearly, if a vertex v_h, $h < m$, receives the announcement, it will select path $R_1[v_h]Q_2$, since the set of choices at v_h is currently empty. Hence, the announcement will be propagated further. This implies that the message will eventually reach vertex $v_{m'}$, $m' > m$.

ii) $\lambda^{v_m}(R_1[v_m]Q_2) > \lambda^{v_m}(R_0[v_m]Q_1)$. We now show that there exists a dispute reel having v_m and u_1 as pivot vertices. Let $\bar{R}$ be the subpath of R_1 from u_1 to v_m, that is, $R_1 = \bar{R}R_1[v_m]$. Now consider the dispute wheel $\Pi' = ((v_m\ u_1), (R_1[v_m]Q_2\ Q_1), (R_0[v_m]\ \bar{R}))$. We now show that Π' is a DR. Being Π a DR, Condition *i* of Definition 3.1 holds since $v_m \notin Q_1$ and $u_1 \notin R_1[v_m]Q_2$. Condition *ii* is trivially satisfied by vertices on paths Q_1 and Q_2, because both are spoke paths in Π. By definition, $\bar{R} \cap R_1[v_m] = \{v_m\}$. Moreover, $R_1[v_m] \cap R_0[v_m] = \{v_m\}$, since, by definition of v_m, $v_j \notin R_1[v_m]$ if $j > m$, and $v_j \notin R_0[v_m]$ if $j < m$. Again, being Π a DR, Condition *iii* holds for paths Q_1 and Q_2, and we have $R_1[v_m] \cap Q_1 = \emptyset$.

We then conclude that if Π is not rim-by-rim, then it contains a dispute reel having two pivot vertices. By Lemma 4.2, instance S is not safe under filtering. ■

By combining Lemmas 4.2, 4.3, and 4.4, we can state the following theorem.

Theorem 4.1: An SPVP instance containing a dispute reel is not safe under filtering.

C. Multiple Solutions and Safety Under Filtering

We now exploit Theorem 4.1 to show that networks admitting multiple stable states are not safe under filtering. Since multiple stable states happen in practice (see, e.g., BGP wedgies [18]), this is especially interesting from an operational perspective.

Theorem 4.2: If an SPVP instance S admits two stable states, then S is not safe under filtering.

Proof: Theorem V.4 in [8] proves that S must contain a dispute wheel Π. Π is derived by merging two stable path assignments π_1 and π_2. Let T_1 and T_2 be the routing trees induced by π_1 and π_2, and let $T = T_1 \cap T_2$. Each spoke path in Π is composed by a path along T plus a final edge which does not connect two vertices in T. Hence, spoke paths form a tree (Condition *iii* of Definition 3.1). Rim paths are built up by vertices which are not in the intersection of π_1 and π_2, thus Condition *ii* is also satisfied. Each pivot vertex u_i can only appear in Q_i, R_i, and R_{i-1} (Condition *i*), since the dispute wheel is built using only $\pi_1(u_i)$ and $\pi_2(u_i)$. Therefore, Π is a dispute reel. By Theorem 4.1, the presence of a dispute reel in S is enough to conclude that S is not SUF. ■

An important consequence of Theorem 4.2 is that observing multiple different stable routing states in a network indicates that its stability may be definitively compromised by the application of route filters. Therefore, the existence of multiple stable states in a network constitutes an important alert to consider for a network operator. As a final remark, we stress that the construction presented in Theorem V.4 of [8] can be exploited to identify a portion of the network which can potentially lead to oscillations under filtering. Moreover, given a set of stable routing states, implementing that construction is straightforward and can be done efficiently. Network operators can use the technique in [8] to disclose a policy dispute in the routing configuration. Our results prove that the presence of such a policy dispute makes the network not SUF.

V. No DR Implies Safety Under Filtering

We now show that the absence of a dispute reel is a sufficient condition for safety under filtering. Combined with the result from the previous section, we can conclude that the presence of a DR characterizes safety under filtering. We prove the sufficient condition by showing that if an SPVP instance is not SUF, then it contains a DR. First, we use the same technique as in [8] to show that a routing oscillation implies the existence of a particular kind of dispute wheel, which satisfies a slightly different set of conditions than those in Definition 3.1. Then, we show that the presence of such a dispute wheel implies the existence of a dispute reel.

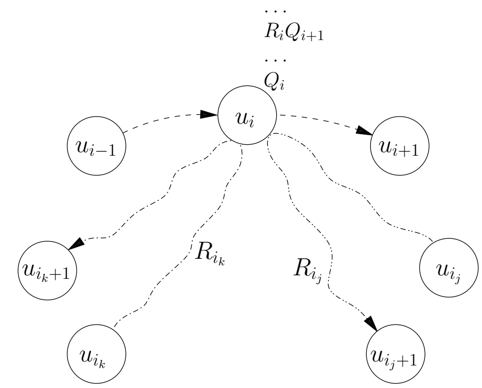


Fig. 10. A dispute wheel where pivot vertex u_i appears in rim paths other than R_i and R_{i-1}. By Lemma 5.2, another dispute wheel can be constructed such that u_i appears in exactly 3 paths.

Lemma 5.1: Consider an SPVP instance S. If S is not safe under filtering, then there exists a dispute wheel Π which satisfies the following conditions:

i) Conditions *ii* and *iii* of Definition 3.1.
ii) For all $u_i \in \vec{\mathcal{U}}$, u_i cannot appear in Q_j, $j \neq i$.
iii) If $u_i \in R_j$, then $R_j[u_i]Q_{j+1}$ is preferred to Q_i.

Proof: Since S is not SUF, there exists a combination of filters inducing an instance S' such that S' is not safe. We can then apply the technique described in Theorem V.9 of [8] to show that S' contains a dispute wheel Π satisfying the above conditions. The statement follows by noting that Π must also be present in S. ■

Observe that the dispute wheel of Lemma 5.1 is not a DR. In particular, it could be the case that a pivot vertex u_i appears in a rim path R_m with $m \notin \{i-1, i\}$. The following lemma shows that such a DW implies the existence of a DR.

Lemma 5.2: Given an instance S, suppose it contains a dispute wheel $\Pi = (\vec{\mathcal{U}}, \vec{\mathcal{Q}}, \vec{\mathcal{R}})$ satisfying the conditions in the statement of Lemma 5.1. Then, S contains a dispute reel.

Proof: If Π is already a DR, the statement trivially holds. Otherwise, for Π not to be a reel, there must exist at least a pivot vertex u_i such that $u_i \in R_m$ with $m \notin \{i-1, i\}$. Let $R_{i_1}, \ldots, R_{i_k}$ be the rim paths traversing u_i, where $i_j \notin \{i-1, i\}$. Without loss of generality, assume that $i_k < i$ is the closest index to i in the order induced by $\vec{\mathcal{U}}$, see Fig. 10. Condition *iii* of Lemma 5.1 ensures that u_i prefers path $R_{i_k}[u_i]Q_{i_k+1}$ to Q_i. Now consider the dispute wheel $\Pi' = (\vec{\mathcal{U}}', \vec{\mathcal{Q}}', \vec{\mathcal{R}}')$, where $\vec{\mathcal{U}}' = (u_i\ u_{i_k+1} \ldots u_{i-1})$, $\vec{\mathcal{Q}}' = (Q_i\ Q_{i_k+1} \ldots Q_{i-1})$, and $\vec{\mathcal{R}}' = (R_{i_k}[u_i]\ R_{i_k+1} \ldots R_{i-1})$. Intuitively, Π' is obtained by "chopping" Π, using path $R_{i_k}[u_i]$ as the new rim path associated with vertex u_i. Observe that every spoke path in Π' is a also spoke path in Π. Moreover, every rim path in Π' except $R_{i_k}[u_i]$ is also a rim path in Π, and $R_{i_k}[u_i]$ is a subpath of R_{i_k}. Therefore, Π' trivially satisfies all the conditions of Lemma 5.1. Moreover, by the definition of index i_k, we know that Π' is such that u_i only appears in Q_i, $R_{i_k}[u_i]$ and R_{i-1}. By applying this construction, we force one pivot vertex at a time to satisfy Condition *i* of Definition 3.1, even if R_{i_k} contains other pivot vertices than u_i. Hence, after iterating the construction at most $|\vec{\mathcal{U}}|$ times, we eventually end up with a dispute reel. ∎

We stress that Condition *iii* of Lemma 5.1 is strictly necessary to apply the construction in Lemma 5.2. As a counterexample, consider again the instance in Fig. 4. In this instance, the DW $((1\ 2\ 3), ((1\ 0)\ (2\ 0)\ (3\ 0)), ((1\ 3\ 2)\ (2\ 1\ 3)\ (3\ 2\ 1)))$ only violates Condition *iii* of Lemma 5.1. In fact, rim path $(1\ 3\ 2)$ traverses pivot vertex 3, but $\lambda^3((3\ 2\ 0)) > \lambda^3((3\ 0))$. It is easy to check that, in this case, no DR can be constructed starting from the DW.

Theorem 5.1: If an SPVP instance S is not safe under filtering, then it contains a dispute reel.

Proof: Lemma 5.1 ensures that S contains a dispute wheel satisfying some particular constraints. We can then apply Lemma 5.2 to find a dispute reel in S. ∎

By combining Theorems 4.1 and 5.1, we conclude that **the absence of a dispute reel is a sufficient and necessary condition for safety under filtering**.

Researchers have deemed the dispute wheel concept important because it only depends on the routing policies. As such, it allows us to prove fundamental properties of the SPVP protocol using just static analysis, i.e., without having to cope with the details of routing dynamics. In fact, the absence of a dispute wheel implies that an SPVP instance is safe under filtering (Corollary 1 of [17]) and has a unique stable state (Theorem V.4 of [8]). Obviously, as safety and robustness can be viewed as special cases of safety under filtering, the absence of a dispute wheel also implies that an SPVP instance is safe and robust. Fig. 3 is a Venn diagram that effectively displays those implications.

As a side effect of our findings, we show that a "no DR" condition can replace the well known "no DW" one in all the above results: in fact, "no DR" is a strictly less constraining condition to show that an SPVP instance is safe, robust, SUF, and has a unique stable state. Moreover, this condition still depends only on the structure of routing policies.

Corollary 5.1: The absence of a DR in an SPVP instance S implies that S has a unique stable state, is safe, and is robust.

Proof: Theorem 4.2 proves that S has a unique stable state. Since safety and robustness are special cases of safety under filtering, Theorem 5.1 proves the rest of the statement. ∎

VI. Safety Under Filtering and Robustness

Safety under filtering is an extremely useful concept to study the impact of route filters on routing stability. An interesting related problem is the impact of link and/or router failures on the safety of BGP. The property of being safe after removing any subset of the vertices or edges from an SPVP instance is referred to as *robustness* [8]. More formally, we say that an instance $S = (G = (V, E), \mathcal{P}, \Lambda)$ is *robust* if every instance $S' = (G' = (V', E'), \mathcal{P}, \Lambda)$ such that $V' \subseteq V$ and $E' \subseteq E$ is safe. Without loss of generality, in the following we only consider link failures.

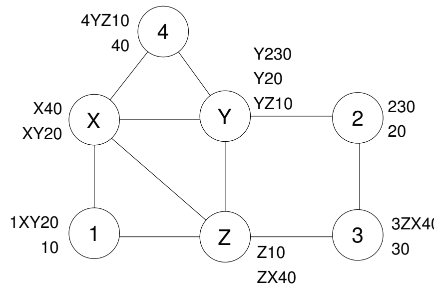


Fig. 11. Filthy-Gadget: an instance which is robust but not safe under filtering. Vertex 0 is omitted for brevity.

As pointed out in [17], the removal of edges and vertices has the same effect as filtering all the paths that traverse those edges and vertices. As a consequence, an instance that is SUF is also robust. Following the findings of Sec. V, we now show that the class of robust SPVP instances is strictly larger than the class of instances that are SUF. Consider the instance Filthy-Gadget in Fig. 11. This instance is clearly not SUF since it contains a DR $\bar{\Pi} = (\bar{\mathcal{U}}, \bar{\mathcal{Q}}, \bar{\mathcal{R}})$, where $\bar{\mathcal{U}} = (1\ 2\ 3\ 4)$, $\bar{\mathcal{Q}} = ((1\ 0)\ (2\ 0)\ (3\ 0)\ (4\ 0))$, and $\bar{\mathcal{R}} = ((1\ X\ Y\ 2)\ (2\ 3)\ (3\ Z\ X\ 4)\ (4\ Y\ Z\ 1))$. Yet, Filthy-Gadget is robust. We prove the latter statement in two parts: first, we show that Filthy-Gadget is safe; second, we show that any combination of link failures produces a safe instance.

To prove the first part, we need the following definition. A vertex v is said to be *prevented from selecting path P* if, for every fair activation sequence, there exists a time t' such that v does not select P (i.e., $\pi_t(v) \neq P$) for any $t > t'$.

Lemma 6.1: Instance Filthy-Gadget is safe.

Proof: Let σ be any fair activation sequence. Given that $\pi_t(0) = (0)$ for all t, by the fairness of σ each neighbor of 0 is prevented from selecting path ϵ. In particular, after some time vertex 2 can only use paths $(2\ 3\ 0)$ or $(2\ 0)$. Since Y accepts both paths from vertex 2, Y is prevented from selecting path $(Y\ Z\ 1\ 0)$, which is less preferred. Vertex 4 is therefore prevented from selecting path $(4\ Y\ Z\ 1\ 0)$. Since 4 is a neighbor of 0, it is also prevented from selecting ϵ. Hence, by the fairness of σ, vertex 4 will end up selecting path $(4\ 0)$ permanently, in turn forcing vertex X to permanently choose path $(X\ 4\ 0)$. Since path $(X\ Y\ 2\ 0)$ will not be advertised by X, vertex 1 is prevented from selecting path $(1\ X\ Y\ 2\ 0)$. Also, being 1 a neighbor of 0, it will end up selecting path $(1\ 0)$ permanently. Vertex Z, in turn, will be forced to select path $(Z\ 1\ 0)$, preventing vertex 3 from selecting $(3\ Z\ X\ 4\ 0)$. By applying the same argument as above, we conclude that vertex 3 will permanently select path $(3\ 0)$. Hence, vertex 2 will select path $(2\ 3\ 0)$, in turn forcing vertex Y to select $(Y\ 2\ 3\ 0)$. It is easy to check that the path assignment induced by σ is stable. Since we did not make any hypothesis on σ, we conclude that Filthy-Gadget is guaranteed to reach this stable path assignment for any fair activation sequences, that is, Filthy-Gadget is safe. ∎

Lemma 6.2: Instance Filthy-Gadget is robust.

Proof: By the previous lemma, we know that Filthy-

GADGET is safe. We now show that any instance S' obtained by removing one or more links from FILTHY-GADGET contains no DR, hence it is safe. Recall that FILTHY-GADGET contains the DR $\bar{\Pi}$ we described above. It is easy to see that its supporting instance $S[\bar{\Pi}]$ is built on the same graph as FILTHY-GADGET. Hence, removing one or more links forcedly creates an instance where $\bar{\Pi}$ does not exist anymore. In order to complete the proof, we need to demonstrate that $\bar{\Pi}$ is the only DR in FILTHY-GADGET. Observe that this is trivially true if vertices X, Y and Z are not pivot vertices. We now show that no DR $\Pi' = (\vec{\mathcal{U}}', \vec{\mathcal{Q}}', \vec{\mathcal{R}}')$ exists having X, Y, or Z as a pivot vertex.

i) Assume that X is a pivot vertex of Π'. Without loss of generality, we say $X = u'_0$. Then $Q'_0 = (X\ Y\ 2\ 0)$ and $R'_0 = (X\ 4)$, which implies $u'_1 = 4$. Since $(Z\ 1\ 0)$ is the best ranked path at vertex Z, we have either $u'_2 = Y$ or $u'_2 = 1$. The former case results in a dispute wheel where spoke path Q'_0 contains a pivot node $u'_2 = Y$. The latter case results in a DW where spoke path Q'_0 shares vertex Y with rim path R'_1. In both cases, Π' cannot be a DR.

ii) We can apply a symmetric argument to vertex Z. Assume that Z is a pivot vertex of Π', $Z = u'_0$. Then $Q'_0 = (Z\ X\ 4\ 0)$ and $R'_0 = (Z\ 1)$, which implies $u'_1 = 1$. As above, if $u'_2 = X$ or $u'_2 = 2$, we find that Π' cannot be a DR. The only other possibility is $u'_2 = Y$, i.e., Y is also a pivot vertex. This case is discussed in the following.

iii) Assume that Y is a pivot vertex of Π'. Without loss of generality, we say $Y = u'_i$. We have two cases, namely $Q'_i = (Y\ Z\ 1\ 0)$ or $Q'_i = (Y\ 2\ 0)$.
- if $Q'_i = (Y\ Z\ 1\ 0)$, then $u'_{i-1} = 4$. We now have either $u'_{i-2} = X$ or $u'_{i-2} = 3$. The former case implies that Q'_{i-2} contains pivot vertex Y. The latter case implies that R'_{i-2} intersects Q'_i at vertex Z. Hence, Π' cannot be a DR.
- if $Q'_i = (Y\ 2\ 0)$, then $u'_{i-1} = 1$. We now have either $u'_{i-2} = Z$ or $u'_{i-2} = 4$. The former case implies that Q'_{i-2} and R'_{i-1} share vertex X. The latter case implies that pivot vertex Y also appears in R'_{i-2}. In both cases, Π' cannot be a DR.

We conclude that $\bar{\Pi}$ is the only DR in FILTHY-GADGET, hence the instance is robust. ■

We performed an exhaustive analysis which independently confirmed the result of Lemma 6.2. Namely, we generated all the possible combinations of failures of one or more links, and then ran the greedy heuristic algorithm in [8] on the resulting instance. That algorithm is correct, that is, it never misreports an instance as safe. Our brute force analysis confirmed that removing one or more links from FILTHY-GADGET results in a safe instance. Unfortunately, the greedy algorithm is not smart enough to also prove the safety of the original instance, because it does not fully exploit vertices that are prevented from selecting specific paths.

VII. CONCLUSIONS

Under the realistic assumption that ASes are allowed to filter routes arbitrarily, the safety of policy-based routing is intrinsically incompatible with unrestricted route rankings. This paper characterizes safety under filtering, determining the amount of autonomy that rankings must sacrifice in order to guarantee stable policy routing. The significance of this result is twofold: on one hand, we fill the large gap that separates currently known necessary and sufficient conditions; on the other hand, we bind safety under filtering to the presence of a particular structure of routing preferences, called dispute reel, which can be statically detected.

An interesting consequence of our results is that a network admitting multiple stable routing states (e.g., BGP wedgies [18]) is not safe under filtering. In this case, we can also pinpoint the problematic portions of the policy configuration.

We finally show that a robust instance may not be safe under filtering. In a sense, this proves that the autonomy of adding (possibly misconfigured) filters can be more harmful than network faults. Finally, as a side effect of our work, we show that the less constraining "no dispute reel" condition can replace the "no dispute wheel" one in a lot of results in the field of policy routing stability.

REFERENCES

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, 2006.
[2] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A measurement study on the impact of routing events on end-to-end Internet performance," in *Proc. SIGCOMM*, 2006.
[3] N. Kushman, S. Kandula, and D. Katabi, "Can you hear me now?! It must be BGP," in *Computer Communication Review*, 2007.
[4] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," *Computer Networks*, vol. 32, no. 1, 2000.
[5] T. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *Proc. SIGCOMM*, 1999.
[6] ——, "A safe path vector protocol," in *Proc. INFOCOM*, 2000.
[7] T. Griffin, F. B. Shepherd, and G. Wilfong, "Policy disputes in path-vector protocols," in *Proc. ICNP*, 1999.
[8] ——, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. on Networking*, vol. 10, no. 2, pp. 232–243, 2002.
[9] L. Gao and J. Rexford, "Stable internet routing without global coordination," in *Proc. SIGMETRICS*, 2000.
[10] L. Gao, T. Griffin, and J. Rexford, "Inherently safe backup routing with BGP," in *Proc. INFOCOM*, 2001.
[11] J. L. Sobrinho, "An algebraic theory of dynamic network routing," *IEEE/ACM Trans. on Networking*, vol. 13, no. 5, pp. 1160–1173, 2005.
[12] ——, "Network routing with path vector protocols: Theory and applications," in *Proc. SIGCOMM*, 2003.
[13] T. Griffin and J. L. Sobrinho, "Metarouting," in *Proc. SIGCOMM*, 2005.
[14] C. Chau, R. Gibbens, and T. Griffin, "Towards a unified theory of policy-based routing," in *Proc. INFOCOM*, 2006.
[15] C. Chau, "Policy-based routing with non-strict preferences," in *Proc. SIGCOMM*, 2006.
[16] N. Feamster, R. Johari, and H. Balakrishnan, "Implications of autonomy for the expressiveness of policy routing," in *Proc. SIGCOMM*, 2005.
[17] ——, "Implications of autonomy for the expressiveness of policy routing," *IEEE/ACM Trans. on Networking*, vol. 15, no. 6, 2007.
[18] T. Griffin and G. Huston, "BGP Wedgies," RFC 4264, Internet Engineering Task Force, Nov 2005.
[19] A. Basu, C. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, "Route oscillations in I-BGP with route reflection," in *Proc. SIGCOMM*, 2002.
[20] L. Cittadini, G. Di Battista, and M. Rimondini, "How stable is stable in interdomain routing: Efficiently detectable oscillation-free configurations," Dept. of Computer Science and Automation, Roma Tre University, Tech. Rep. RT-DIA-132-2008, Jul 2008.