# Principles of Safe Policy Routing Dynamics

SAMUEL EPSTEIN KARIM MATTAR IBRAHIM MATTA
Computer Science Department, Boston University
{samepst, kmattar, matta}@cs.bu.edu

***Abstract*—We introduce the *Dynamic Policy Routing* (DPR) model that captures the propagation of route updates under arbitrary changes in topology or path preferences. DPR introduces the notion of *causation chains* where the route flap at one node causes a flap at the next node along the chain.**

**Using DPR, we model the Gao-Rexford (economic) guidelines that guarantee the safety (*i.e.,* convergence) of policy routing. We establish three principles of safe policy routing dynamics. The *non-interference principle* provides insight into which ASes can directly induce route changes in one another. The *single cycle principle* and the *multi-tiered cycle principle* provide insight into how cycles of routing updates can manifest in any network.**

**We develop INTERFERENCEBEAT, a distributed algorithm that propagates a small token along causation chains to check adherence to these principles. To enhance the diagnosis power of INTERFERENCEBEAT, we model four violations of the Gao-Rexford guidelines (*e.g.,* transiting between peers) and characterize the resulting dynamics.**

## I. INTRODUCTION

The Border Gateway Protocol (BGP) is currently the defacto inter-domain routing protocol employed in the Internet. BGP allows Autonomous Systems (ASes), operated by different administrative domains (*e.g.,* Internet Service Providers, companies, universities) to independently apply local policies for selecting routes and propagating routing information. Given the critical role and global scope of BGP, both its transient and steady-state performance have received significant attention, and problems related to delayed convergence [1] and potential instability [2], [3] (*i.e.*, route oscillations/flaps) have been identified and studied.

Route flaps in particular can be highly disruptive given the associated cost of communication and processing overheads. Route flaps can be transient (*i.e.*, short-term) due to temporary changes in topology or route/path preferences. Route flaps can also be persistent due to conflicting routing policies across ASes (*i.e.*, policies can not be simultaneously satisfied) [4].

Economic constraints that are typical of commercial relationships between ASes in the Internet—henceforth referred to as the Gao-Rexford guidelines [5]—have been shown to make BGP free from policy conflicts (*i.e.,* convergent). We refer to routing policy instances that adhere to the Gao-Rexford guidelines as *safe* and ones that do not as *potentially unsafe*. The Gao-Rexford guidelines are:

1) An AS classifies its neighboring ASes as either customer, peer or provider.
2) The path preferences are restricted in a hierarchical fashion. Every AS prefers a path through a customer AS over a path through a peer/provider AS.
3) All advertised paths are "valley-free". They consist of zero or more customer-to-provider links followed by an optional peering link followed by zero or more provider-to-customer links.

*Our Contribution:*

We extend the Stable Paths Problem [4] (a static model of BGP) to capture the propagation dynamics of route updates under arbitrary changes in topology (*e.g.,* link failures) or path preferences (*e.g.,* policy configuration updates). We call this extended model the *Dynamic Policy Routing* (DPR) model. DPR introduces the notion of a *causation chain* which is informally defined as a sequence of nodes where the route flap at one node causes a flap at the next node along the chain.

We model a strict version of the Gao-Rexford guidelines which we call the *economic DPR* model. We prove the existence of several invariant properties of causation chains irrespective of arbitrary changes in topology or path preferences. For example, we prove that all causation chains in the economic DPR model are valley-free, thus generalizing the result in [6] to dynamic networks. Violations of the economic DPR model result in potentially unsafe routing behavior where the causation chains are not necessarily valley-free.

We develop INTERFERENCEBEAT, a distributed algorithm that checks if the routing dynamics adhere to the ones predicted by the economic DPR model. If not, then the presence of policy violations can be inferred. INTERFERENCEBEAT appends a token to each routing update message. Tokens are propagated along causation chains.

We model four common policy violations (*e.g.,* transiting between peers). For each violation, we prove the invariant properties of the resulting causation chains. Using these inferred properties, we extend the diagnosis power of INTERFERENCEBEAT. The novelty of this work is that:

1) We identify key principles (*i.e.,* invariant properties) of safe policy routing dynamics regardless of changes to the underlying topology or path preferences.
2) We identify and model four common violations of safe policy routing and characterize the resulting dynamics.
3) We introduce INTERFERENCEBEAT, a distributed algorithm to detect and diagnose policy violations.

## II. PRINCIPLES OF SAFE POLICY ROUTING DYNAMICS

In this section, we distill the key results of our DPR model into three principles. These principles capture several useful invariant properties of the routing dynamics under safe policy routing (*i.e.,* where the policies of all nodes adhere to the Gao-Rexford guidelines). We discuss reasons why ASes violate these guidelines leading to potentially unsafe dynamics where these principles no longer hold.

We also show that routing dynamics need to be explicitly considered when detecting policy violations. We postpone formal definitions to later sections and focus here on presenting the main intuitions behind our results.

### A. What are the principles?

**Non-Interference Principle:** *If an AS $y$ is not at a higher tier-level than (provider to) any two of its neighbors $x$ and $z$, then $x$ and $z$ cannot directly induce path changes in each other through $y$. This principle holds regardless of changes in the underlying topology or path preferences.*

The notion of "inducing path changes" is synonymous with a continuous propagation of path changes across nodes, which we model in DPR as a causation chain. The basic premise of the non-interference principle comes from a result in DPR (Theorem 1 in Section IV) where we proved that any causation chain must not contain sequences such as a provider-to-customer-to-provider.

Figure 1 outlines all the Internet configurations where AS $x$ cannot directly affect AS $z$ through AS $y$. More specifically, non-interference holds if:

1) AS $y$ is multi-homed with providers AS $x$ and AS $z$.
2) AS $y$ is a customer of AS $x$ and a peer of AS $z$.
3) AS $y$ is a peer of AS $x$ and a customer of AS $z$.
4) AS $y$ is a peer of both AS $x$ and AS $z$

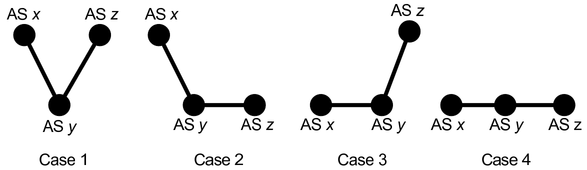


Fig. 1. All Internet configurations where AS $x$ cannot directly affect AS $z$. Horizontal edges represent peering links and diagonal edges represent customer-to-provider links.

**Single Cycle Principle:** *In any cycle of routing update messages between ASes, every AS $x$ affects its neighbor $y$ at most once. This principle holds regardless of changes in the underlying topology or path preferences.*

The notion of "cycle" is synonymous with a continuous propagation of path changes across nodes where at least one node is affected twice. We model such a cycle of path changes in DPR as a *causation cycle*. The single cycle principle comes from a result in DPR (Theorem 2 in Section IV) where we proved that any causation cycle in safe policy routing occurs only once.

**Multi-Tiered Cycle Principle:** *Every cycle of routing update messages between ASes must have at least two ASes in different tier-levels. This principle holds regardless of changes in the underlying topology or path preferences.*

The multi-tiered cycle principle comes from a result in DPR (Theorem 2 in Section IV) where we proved that no causation cycle in safe policy routing can occur exclusively between peering ASes.

### B. Why do the principles not always hold?

Violations of safe policy routing (*i.e.,* the Gao-Rexford guidelines) result in unpredictable, black-box dynamics that are potentially unsafe. When policy violations occur, the principles no longer hold (Table III in Section VI). The reasons for such violations are:

1) **Intentional:** representing legitimate policy configurations for backup links or complex agreements [7].
2) **Unintentional:** representing misconfigurations or complex real-time interactions between routers that do not reflect the intentions of the administrators.

### C. How do we check the principles?

Network administrators can *locally* check whether they are conforming to the Gao-Rexford guidelines where the dynamics are guaranteed to conform to the principles. This can be done by inspecting their local preferences and ensuring that all adopted paths are valley-free.

Local checks are inadequate since not all nodes are necessarily compliant with the guidelines. Figure 2 illustrates "interference" between nodes 1 and 3. The interference is due to policy violations by node 2 which cannot be locally checked by node 3. Instead, node 3 will need to discover the interference by somehow detecting the causation chain propagating through nodes 1, 2 and 3.

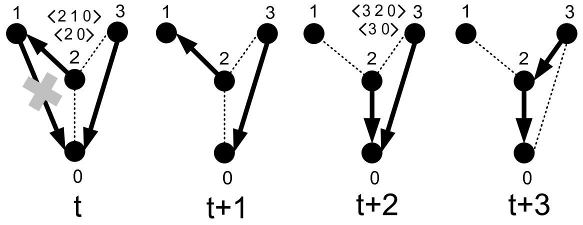


Fig. 2. Sample dynamics where interference occurs. The list of path preferences for nodes 2 and 3 are organized such that the most preferred path is at the top. Paths not explicitly listed are forbidden. All nodes are trying to reach destination node 0.

Node 3 is abiding by the Gao-Rexford guidelines and initially uses the customer path $\langle 30 \rangle$ which is valley-free. Node 2, however, violates the guidelines by preferring a path through its provider $\langle 210 \rangle$ over a path through its customer $\langle 20 \rangle$. At time $t$, the link connecting node 1 to node 0 is lost, causing node 1 to have an empty path to node 0 at time $t+1$. At time $t+2$, node 2 switches from path $\langle 210 \rangle$ to $\langle 20 \rangle$. This action in turn causes node 3 to switch from path $\langle 30 \rangle$ to $\langle 320 \rangle$ at time $t{+}3$. Even though node 3 abides by the Gao-Rexford guidelines, the forbidden interference occurs. The causation chain consists of a provider (node 1), followed by its customer (node 2), followed by another provider (node 3).

If node 2 does not violate the guidelines, the dynamics would manifest differently. Suppose the path $\langle 20 \rangle$ is forbidden, *forcing* node 2 to use its provider path $\langle 210 \rangle$. The loss of link connectivity between nodes 1 and 0 at time $t$ causes node 2 to lose connectivity at time $t+2$. Node 3 is unaffected. The causation chain solely consists of a provider (node 1) followed by its customer (node 2). Since this chain is valley-free, the dynamics conform to the principles.

## III. Dynamic Policy Routing Model

The Dynamic Policy Routing (DPR) model is used to capture the dynamics of BGP. Each AS is represented by a node in a graph. AS path preferences are represented by a ranking relation. DPR extends SPP [4] to model time-varying topologies and path preferences. The central notions in DPR are that of *action* and *causation*. An action corresponds to a routing decision made upon the reception of a routing update message. A causing node corresponds to the node sending that update message. DPR models these two events to construct a *causation chain* over time where each node causes its successor along the chain to take an action.

### A. Basics of DPR

**Definition 1** (Time)**.** Time is represented by a non-negative, discrete index $t$ such that: $t \in [0, \infty)$.

**Definition 2** (Network)**.** The network is represented by a graph $G = (V, E)$:

- Each vertex $u \in V$ represents an AS.
- Each edge in $E$ is time dependent: $(u, v)^t \in E$ if $u$ is connected to $v$ at time $t$. Conversely, a lack of connectivity between $u$ and $v$ at time $t$ (*i.e.,* link failure) is represented by $(u, v)^t \notin E$.

There exists a distinguished destination node, represented as *root*, where *root* $\in V$. In other words, DPR considers a single destination prefix.

**Definition 3** (Paths)**.** Paths are sequences of nodes of the form: $\langle u_1\ u_2\ \ldots\ u_k \rangle$ where the destination node *root* is $u_k$. The empty path is denoted by $\langle\rangle$. A concatenation of a node $u$ with a path $Q$ is represented as: $P = \langle u\, Q \rangle$. A path originating from $u$ is represented by $P^u$. The set of paths originating from $u$ is represented by $\mathcal{P}^u$.

**Definition 4** (Path Preferences)**.** At each time $t$, each node $u$ has a unique preference over paths originating at $u$. This dynamic ranking is represented by the $\succeq^t$ operator. If $u$ prefers $P^u$ over $Q^u$ at time $t$ then: $P^u \succeq^t Q^u$. If $u$ prefers $P^u$ over $Q^u$ for all $t$ then: $P^u \succeq Q^u$. Strict preference is defined by:

$$P^u \succ^t Q^u \text{ iff } P^u \succeq^t Q^u \text{ and } Q^u \not\succeq^t P^u$$

For all times $t$, for each node $u \in V$, $\succeq^t$ is a total order over $\mathcal{P}^u \cup \langle\rangle$. Thus each node $u$ has an ordered preference over all its paths to *root*. If two paths start with different nodes, then they have no preference relation. Forbidden paths $P$ are those ranked below the empty path for all times: $\langle\rangle \succ P$. All paths with repeating nodes are forbidden.

**Definition 5** (DPR Instance)**.** A Dynamic Policy Routing (DPR) instance consists of a graph and a path preference $D = (\succeq^t, G)$.

**Definition 6** (Best Paths)**.** At each time index $t$, every node $u$ has a path to *root*, represented by $P^u = \pi(u, t)$. The available path choices of a node, via all possible neighbors $v$, are represented by $\text{Choices}(u, t)$ where:

$$\text{Choices}(u, t) = \langle\rangle \cup \{\langle u\ \pi(v, t)\rangle ; (u, v)^t \in E\}$$

The $\text{Best}(u, t)$ notation represents the current best path for $u$:

$$\text{Best}(u, t) = \max_{\succeq t}\ \text{Choices}(u, t)$$

The paths assigned to nodes at each time $t$ is their best path of the previous round. For all nodes $u \in V$:

- $\pi(u, 0) = \langle\rangle$
- $\pi(u, t) = \text{Best}(u, t-1)$

The path used by node $u$ at time $t$, $\pi(u, t)$, was its best path at time $t-1$, $\text{Best}(u, t-1)$. This best path was determined using the ranking $\succeq^{t-1}$.

**Definition 7** (Next-Hop Neighbor)**.** The $\rho$ notation is used to represent the next-hop neighbor of a current path:

$$\rho(u, t) = \text{NextHop}(\pi(u, t))$$

**Definition 8** (Realized Paths)**.** A path $P^u$ is *realized* iff there exists a time $t$ such that $\pi(u, t) = P^u$.

**Proposition 1** (Path Deconstruction)**.** *If* $\rho(u_0, t) = u_1$ *then* $\pi(u_0, t) = \langle u_0\ \pi(u_1, t-1)\rangle$

*Proof:* By the definition of $\pi$, $\pi(u_0, t) = \text{Best}(u_0, t-1)$ so $\pi(u_0, t) \in \text{Choices}(u_0, t-1)$. So by the definition of Choices, $\pi(u_0, t) = \langle u_0\ \pi(u_1, t-1)\rangle$, where $u_1 = \rho(u_0, t)$. ■

### B. Causation in DPR

**Definition 9** (Path Rank Changes)**.** The following definitions describe the relative change in the rankings of selected paths for a node:

$$\begin{aligned} \text{RankDec}(u, t) &\quad \text{iff} \quad \pi(u, t) \succ^t \pi(u, t+1) \\ \text{RankInc}(u, t) &\quad \text{iff} \quad \pi(u, t) \prec^t \pi(u, t+1) \\ \text{RankSame}(u, t) &\quad \text{iff} \quad \pi(u, t) = \pi(u, t+1) \end{aligned}$$

The relative change in rankings are with respect to the current path ranking $\succeq^t$.

**Definition 10** (Causation Function)**.** In DPR, a node $u$ may change its current path at a given time $t$. The causation function represents $u$'s neighboring node $v$ responsible for $u$'s path change. Causation function is the base construct from which causation chains will be built. A causation function $C$ maps each node $u$ at a given time $t$ to a neighboring node $v$: $C(u, t) = v$.

The operating conditions for the causation function are outlined in Table I. There are three cases for the causation function $C(u, t) = v$:

1) Node $v$ was the next hop of $u$'s chosen path at time $t$. However, node $v$ changed its path at time $t$, causing $u$ to choose a less preferred path at time $t+1$.
2) Node $v$ advertised a new path at time $t$, causing $u$ to choose a more preferred path through $v$ at time $t+1$.
3) $v$ is empty, because $u$'s path did not change between times $t$ and $t+1$.

TABLE I
CAUSATION FUNCTION

| | | |
|---|---|---|
| Condition 1: RankDec$(u,t)$ | $\Rightarrow$ | $C(u,t) = \rho(u,t)$ |
| Condition 2: RankInc$(u,t)$ | $\Rightarrow$ | $C(u,t) = \rho(u,t+1)$ |
| Condition 3: RankSame$(u,t)$ | $\Rightarrow$ | $C(u,t)$ is empty |

**Definition 11** (Causation Chain)**.** A causation chain is a sequence of nodes where each node $y_{i-1}$ causes $y_i$ to change its current path. It is represented by $Y = \langle y_0\ y_1\ \dots\ y_k\rangle^t$, where $C(y_i, t+i) = y_{i-1}$, for all $0 < i \leq k$. Time $t$ is defined with respect to $y_0$, and it takes $i$ time steps to build the causation chain up to node $y_i$. An example of a causation chain can be seen in Figure 3.

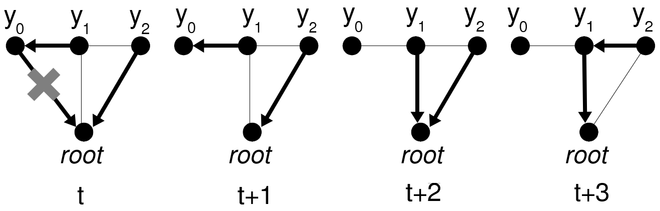


Fig. 3. Causation chain $Y = \langle y_0\ y_1\ y_2\rangle^t$. A link failure between $y_0$ and *root* occurred at time $t$, causing $y_0$ to have no path to root at time $t+1$. This causes $y_1$ to switch to a less preferred path at time $t+2$, where $C(y_1, t+1) = y_0$ with causation condition 1. This causes $y_2$ to switch to a more preferred path via $y_1$ at time $t+3$, where $C(y_2, t+2) = y_1$ with causation condition 2.

**Definition 12** (Causation Cycle)**.** Given a causation chain $\langle y_0\ y_1\ \dots\ y_k\ y_{k+1}\rangle^t$, if $y_0 = y_k$ then a causation cycle $\langle y_0\ y_1\ \dots\ y_k\rangle^t$ exists. If $y_1 \neq y_{k+1}$, then the cycle is *simple*, otherwise the cycle is *non-simple*. The following causation chains contain simple and non-simple cycles:

$$\begin{array}{ll} \text{Simple:} & \langle y_0\ y_1\ y_2\ y_0\ y_3\rangle^t \\ \text{Non-Simple:} & \langle y_0\ y_1\ y_2\ y_0\ y_1\rangle^t \end{array}$$

## IV. ECONOMIC DPR MODEL

This section will show that if a DPR instance conforms to a strict version of the Gao-Rexford guidelines [5], then its dynamic behavior can be characterized, regardless of changes in topology or path preferences. In particular, we show that all causation chains have the property known as "valley-free" and all causation cycles are simple. The economic constraints we consider are as follows:

1) Every node is customer, peer, or provider to its neighboring nodes. The commercial agreement (*i.e.,* relationship) between any two nodes does not change over time.
2) A node cannot be a provider to itself. There are no customer-provider cycles. Furthermore, a node cannot be both a (direct or indirect) provider and a (direct or indirect) peer to another node.
3) For all times, each node prefers a path through a customer over a path through a peer/provider and prefers a path through a peer over a path through a provider.
4) Each node provides transit service only to its customers. Thus, all paths are valley-free.

These economic constraints are a stricter version of the Gao-Rexford guidelines which are sufficient to guarantee stability in a static graph. Thus, the economic DPR model is safe. The restrictions of the economic model enable equivalence classes of peers, as seen in Figure 4.

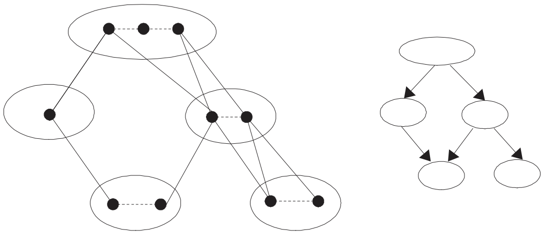


Fig. 4. Equivalence classes of peers in economic DPR.

### A. Basics of Economic DPR

**Definition 13** (Economic Operator)**.** The economic relationship between nodes are described using the operator $\succeq_\$$. This operator is essential for reasoning about the economic relationships between nodes in both paths and causation chains. A *strict* economic relation is defined by:

$$u \succ_\$ v \text{ iff } u \succeq_\$ v \text{ and } u \not\preceq_\$ v$$

and an equivalence relation is defined by:

$$u =_\$ v \text{ iff } u \succeq_\$ v \text{ and } u \preceq_\$ v$$

Economic relationships can be derived from the operator $\succeq_\$$:

- If $u$ is a customer of $v$ then $u \prec_\$ v$.
- If $u$ is a provider to $v$ then $u \succ_\$ v$.
- If $u$ is a peer to $v$ then $u =_\$ v$.

The properties of the economic operator $\succeq_\$$ can be modelled using *pre-order* conditions [8]:

1) (reflexive) $x \succeq_\$ x$
2) (transitive) $x \succeq_\$ y$ and $y \succeq_\$ z$ implies $x \succeq_\$ z$

The following transitive relationships hold:

$$x \succ_\$ y \text{ and } y \succeq_\$ z \text{ implies } x \succ_\$ z$$

$$x \succeq_\$ y \text{ and } y \succ_\$ z \text{ implies } x \succ_\$ z$$

**Definition 14** (Customer, Peer, and Provider Paths)**.** We define paths by the economic relationship between a path's starting node $u$ and its next-hop. For all paths $P^u$:

$$\begin{array}{rll} \text{Customer}(P^u) & \text{iff} & u \succ_\$ \text{NextHop}(P^u) \\ \text{Peer}(P^u) & \text{iff} & u =_\$ \text{NextHop}(P^u) \\ \text{Provider}(P^u) & \text{iff} & u \prec_\$ \text{NextHop}(P^u) \end{array}$$

**Definition 15** (Valley)**.** We define a valley to be a sequence of three distinct nodes $\langle a\ b\ c\rangle$ satisfying the condition:

$$a \succeq_{\$} b \preceq_{\$} c$$

The four types of valleys can be seen in Figure 5. Every valley-free sequence is a series of zero or more ascending customer-to-provider relationships, followed by an optional peer relationship, followed by a series of zero or more descending provider-to-customer relationships.

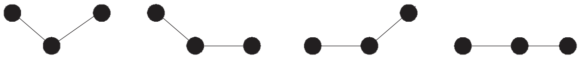


Fig. 5. Valleys

**Definition 16** (Economic DPR Instances)**.** An economic DPR instance $(\succeq_{\$}, \succeq^t, G)$ satisfies the following conditions:

1) All paths which have a valley are forbidden.
2) Customer paths are always preferred over peer/provider paths and peer paths are always preferred over provider paths. Thus given paths $P^u_1$ and $P^u_2$:

$$\begin{array}{lcl} \text{Customer}(P^u_1) \text{ and not Customer}(P^u_2) & \Rightarrow & P^u_1 \succ P^u_2 \\ \text{Peer}(P^u_1) \text{ and Provider}(P^u_2) & \Rightarrow & P^u_1 \succ P^u_2 \end{array}$$

### *B. Causation in Economic DPR*

This section characterizes causation chains and cycles for economic DPR instances.

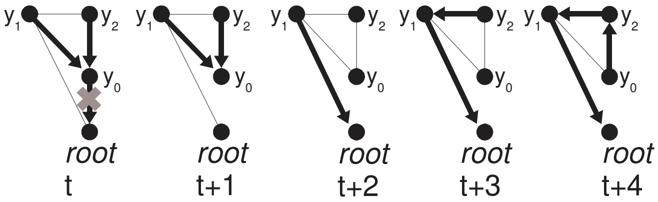


Fig. 6. Causation cycle $Y = \langle y_0\ y_1\ y_2\ y_0\rangle^t$. A link failure between $y_0$ and *root* occurred at time $t$, causing $y_0$ to have no path to root at time $t+1$. This causes $y_1$ to switch to a less preferred path at time $t+2$, where $C(y_1, t+1) = y_0$ with causation condition 1. This causes $y_2$ to switch to a path through $y_1$ at time $t+3$, where $C(y_2, t+2) = y_1$ with causation condition 2. The cycle is closed with $y_0$ switching to a path via $y_2$ at time $t+4$, where $C(y_0, t+3) = y_2$ with causation condition 2. Note the existence of a separate causation chain $Y' = \langle y_0\ y_2\rangle^t$.

**Theorem 1.** *Every causation chain of an economic DPR instance $(\succeq_{\$}, \succeq^t, G)$ is valley-free.*

For ease of exposition, the full proof of Theorem 1 is in Appendix A. In the proof, we assume that there exists a causation chain that has a valley consisting of three consecutive nodes $\langle a\ b\ c\rangle^t$. First we prove that at no time during the causation chain did $b$ have a customer path. Then we prove that at some time during the causation chain, $c$ had a path through $b$. Since $b$ is a customer/peer to $c$ and $b$ does not have a customer path then $c$ had a realized valley path through $b$, causing a contradiction.

**Definition 17** (Horizontal Cycle)**.** A causation cycle is horizontal if all adjacent nodes in the cycle are peers.

**Definition 18** (Vertical Cycle)**.** A causation cycle is vertical if there is at least one customer/provider relationship between adjacent nodes in the cycle.

Figure 6 represents a simple vertical causation cycle, where node $y_0$ loses a path to *root* and reroutes through $y_2$.

**Lemma 1.** *Given a causation cycle $Y = \langle y_0\ \ldots\ y_k\rangle^t$ of an economic DPR instance $(\succeq_{\$}, \succeq^t, G)$, every node in $Y$ is a provider to the first node $y_0$.*

*Proof:* Let $y_i \in Y$, where $0 < i < k$. By Theorem 1, $Y$ is valley-free and either $y_{i-1} \preceq_{\$} y_i$ or $y_i \succeq_{\$} y_{i+1}$. If the first case is true, then by the definition of valley-free paths $y_{j-1} \prec_{\$} y_j$ for all $0 < j < i$, and by the transitive nature of economic relationships, $y_0 \prec_{\$} y_i$. If the second case is true, then by the definition of valley-free paths $y_j \succ_{\$} y_{j+1}$ for all $i < j < k$, and by the transitive nature of economic relationships, $y_i \succ_{\$} y_k$. Thus every node $y_i$ is a provider to $y_0 = y_k$. ■

**Theorem 2.** *Every causation cycle $Y = \langle y_0\ \ldots\ y_k\rangle^t$ of an economic DPR instance is vertical and simple.*

*Proof:* Lemma 1 directly implies that every causation cycle in economic DPR instances are vertical. The second part regarding simple causation cycles is proved by contradiction. Assume there exists a non-simple causation cycle $Y_1 = \langle y_0\ y_1\ \ldots\ y_k\ y_1\rangle^t$ where $y_0 = y_k$. From Lemma 1, $y_0 \prec_{\$} y_1$. However a new causation cycle $Y_2$ exists where: $Y_2 = \langle y_1\ y_2\ \ldots\ y_{k-1}\ y_k\ y_1\rangle^{t+1}$. Thus by Lemma 1, $y_1 \prec_{\$} y_k = y_0$ which is a contradiction. ■

The theoretical results in this section are the proofs for the three principles of safe policy routing dynamics introduced in Section II. The non-interference principle comes from Theorem 1, which states that every causation chain in an economic DPR instance must be valley-free. The single and multi-tiered cycle principles come from Theorem 2, which states that every causation cycle in an economic DPR instance is vertical and simple.

## V. InterferenceBeat

In this section, we outline a distributed algorithm, InterferenceBeat, that checks if the principles of safe policy routing dynamics are maintained or whether policy violations exist. This is accomplished by detecting forbidden causation chains (including cycles) induced by policy violations. Once a forbidden causation chain is detected, the ASes involved need to collaborate to resolve the potential problem.

### *A. Description of* InterferenceBeat

InterferenceBeat piggybacks a small token alongside route updates. When a node $y$ receives a route update from its neighbor $v$ at time $t$, it also receives a token $\theta_{\text{in}}$. If node $y$ selects a new path then it broadcasts a new token $\theta_{\text{out}}$ alongside its own route update at time $t+1$. Tokens are passed along causation chains. In general, a causation chain is started when a link flaps (*i.e.,* is lost or becomes available) or when a node changes its path preferences. A token consists of three parts, $(i, r, n)$. The identifier of the causation chain is $i$. The economic relationship between $y$ and its predecessor $v$ on the causation chain is $r \in \{\succ_{\$},\ \prec_{\$}$ $,\ =_{\$},\ \emptyset\}$. For example, if $v$ is a provider to $y$, then $r$ is $\succ_{\$}$. The counter $n$ keeps track of the number of times the token was passed along a customer-to-provider or a provider-to-customer link.

The PROCESS function outlined in Figure 7 performs basic routing tasks and handles the incoming and outgoing tokens. It is invoked in every node $y$ at time $t$ after receiving all routing update messages. In steps 2 and 3, node $y$ chooses and adopts its best available path. If $y$'s assigned path has changed in step 4 (*i.e.,* an action occurred), then node $y$'s causing neighbor $v$ is identified in step 5. The token received from neighbor $v$ is recovered in step 6. In step 7, the CREATETOKEN function is called which returns the contents of the new token to be sent out by $y$ at time $t+1$. The CHECKPRINCIPLES function is called in step 8. Node $y$ stores information about the outgoing token in step 9, which is later used to detect cycles in the CHECKPRINCIPLES function. The outgoing token is then disseminated to all $y$'s neighbors in step 10.

```
 1: function PROCESS($y, t$)
 2:     Best($y, t$) $\leftarrow \max_{\succeq^t}$ Choices($y, t$)
 3:     $\pi(y, t+1) \leftarrow$ Best($y, t$)
 4:     if $\pi(y, t+1) \neq \pi(y, t)$ then
 5:         $v = C(y, t)$
 6:         $\theta_{\text{in}}$ =GETTOKENFROMNEIGHBOR($y, v, t$)
 7:         $\theta_{\text{out}}$ = CREATETOKEN($y, v, \theta_{\text{in}}$)
 8:         CHECKPRINCIPLES($y, v, \theta_{\text{in}}, \theta_{\text{out}}$)
 9:         STORETOKEN($y, v, \theta_{\text{out}}$)
10:         SENDTOKEN($y, t, \theta_{\text{out}}$)
```

Fig. 7. PROCESS function.

The CREATETOKEN function is outlined in Figure 8. Step 2 retrieves the needed parts from the incoming token. If the identifier $i_{\text{in}}$ is empty in step 3 then a new one is generated in step 4. Otherwise, in step 6, the outgoing identifier $i_{\text{out}}$ is set to the incoming identifier $i_{\text{in}}$. In step 7, $r_{\text{out}}$ is set to the economic relationship between $v$ and $y$. In steps 8 through 11, the outgoing counter $n_{\text{out}}$ is only incremented if nodes $y$ and $v$ are *not* peers. The outgoing token is returned in step 12.

The CHECKPRINCIPLES function is outlined in Figure 9. Steps 2 and 3 retrieve the needed parts from the tokens. Step 4 checks for the existence of a valley causation chain. If one is found, then interference is reported, where the causing node $v$, the chain identifier $i_{\text{in}}$ and the relationship $r_{\text{in}}$ are identified. In step 6, node $y$ determines if it has previously received a token with identifier $i_{\text{in}}$. If so, then a cycle is detected. Node $y$ recovers the old information in step 7. If the token was previously received from the same neighbor $v$ then a non-simple cycle is reported in step 9. Step 10 checks if the token previously received contained the same counter value. If so, then the token was only passed between peers since leaving node $y$ and a horizontal cycle is reported in step 11.

```
 1: function CREATETOKEN($y, v, \theta_{\text{in}}$)
 2:     $(i_{\text{in}}, \_, n_{\text{in}}) = \theta_{\text{in}}$
 3:     if $i_{\text{in}}$ is $\emptyset$ then
 4:         $(i_{\text{out}}, r_{\text{out}}, n_{\text{out}})$ =(NEWID(), $\emptyset$, 0)
 5:     else
 6:         $i_{\text{out}} = i_{\text{in}}$
 7:         $r_{\text{out}}$ = ECONOMICRELATION($v$, $y$)
 8:         if $r_{\text{out}}$ is equal to $=_{\$}$ then
 9:             $n_{\text{out}} = n_{\text{in}}$
10:         else
11:             $n_{\text{out}} = n_{\text{in}} + 1$
12:     return $(i_{\text{out}}, r_{\text{out}}, n_{\text{out}})$
```

Fig. 8. CREATETOKEN function.

```
 1: function CHECKPRINCIPLES($y, v, \theta_{\text{in}}, \theta_{\text{out}}$)
 2:     $(i_{\text{in}}, r_{\text{in}}, \_) = \theta_{\text{in}}$
 3:     $(\_, \_, n_{\text{out}}) = \theta_{\text{out}}$
 4:     if ($r_{\text{in}}$ is equal to $\succ_{\$}$ or $=_{\$}$) and ($v \preceq_{\$} y$) then
 5:         REPORTINTERFERENCE($y, v, \theta_{\text{in}}$)
 6:     if HASRECEIVEDTOKEN($y, i_{\text{in}}$) then
 7:         $(v_{\text{old}}, n_{\text{old}})$ = GETSTOREDTOKEN($y, i_{\text{in}}$)
 8:         if $v_{\text{old}}$ is equal to $v$ then
 9:             REPORTNONSIMPLECYCLE($y, v, \theta_{\text{in}}$)
10:         if $n_{\text{old}}$ is equal to $n_{\text{out}}$ then
11:             REPORTHORIZONTALCYCLE($y, v, \theta_{\text{in}}$)
```

Fig. 9. CHECKPRINCIPLES function.

### B. *Sample Operation of* INTERFERENCEBEAT

Figure 10 shows the operation of INTERFERENCEBEAT on the DPR instance described in Figure 3, assuming $y_0$, $y_1$ and $y_2$ are all peers. At time $t+1$, node $y_0$ initiates a new causation chain with identifier ID1 and sends a token to $y_1$. Since $y_0$ initiated the chain, the count is 0 and the relationship is $\emptyset$. Node $y_1$ takes an action and sends a new token to $y_2$. Since $y_1$ and $y_0$ are peers, the relationship is set to $=_{\$}$ and the count is still 0 as the token only traversed a peering link. Finally, since $y_2$ is a peer to its causing node $y_1$, interference is detected by $y_2$ upon receiving the token.

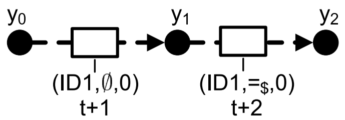


Fig. 10. Sample operation of INTERFERENCEBEAT.

### C. *Properties of* InterferenceBeat

InterferenceBeat has the following characteristics:

- **Efficient Space**. A small token of space complexity $O(1)$ (a few bytes) is appended to each routing update message irrespective of how the routing dynamics manifest in the network.
- **Provably Correct**. InterferenceBeat is based on a comprehensive theory of policy routing dynamics and hence is provably correct with any dynamic network. In other words, any changes in network topology or path preferences do not affect the correctness of detecting policy violations.
- **Incrementally Deployable**. InterferenceBeat requires only a *minor* modification to BGP and can be deployed incrementally. To detect policy violations, only the ASes along the causation chains to be diagnosed need to adopt the protocol. Thus neighboring ASes can use InterferenceBeat to detect misconfigurations.
- **Privacy Preserving**. ASes do not reveal local policy information and only AS relationships are explicitly shared.

### D. *Practical Considerations for* InterferenceBeat

InterferenceBeat could be implemented over BGP where the token is passed in the message options. When an AS initiates a new causation chain it must create a new identifier using the NewId() function. This can be accomplished by hashing the AS number, router identifier, time and destination prefix. A fixed number of bits can be allocated to the identifier, with more bits reducing the probability of a hash collision.

In InterferenceBeat, if a cycle or valley is detected by a node $y$, only its causing neighbor node $v$ can be immediately identified. In order to identify/notify other nodes along the chain, a back-propagating alert protocol may be used. Each node can leverage its stored tokens to find its previous causing neighbor. Note that a token only needs to be stored for the duration of the causation chain, thus the local storage requirements at a node are expected to be minimal.

In [9] we show that the synchronicity of DPR is not a hindrance and that it has sufficient expressive power to model asynchronicity. Hence, InterferenceBeat can be trivially extended to a real-time setting.

## VI. Violations of the Economic DPR Model

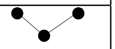
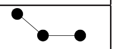
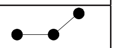
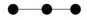
We formally define four common policy violations, which represent different relaxations to the strict economic DPR model[1]. For each violation we prove the invariant properties of the resultant causation chains and cycles. The modelled dynamics induced by each violation can be compared against the dynamics observed by InterferenceBeat. If a violation cannot cause the observed behavior, then it can be ruled out.

### A. *Description of Violations*

To describe paths and causation chains in better detail we categorize valleys into four subtypes.

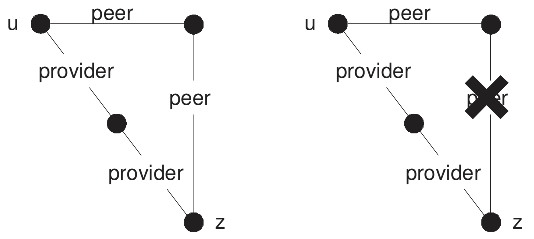
**Definition 19** (Valley Types)**.** We extend definition 15 of valleys to four subtypes as shown in Table II.

TABLE II
Valley types given sequence $\langle a\ b\ c\rangle$.

| Valley Type | Condition | Illustration |
|---|---|---|
| $\mathcal{A}$ | $a \succ_\$ b \prec_\$ c$ |  |
| $\mathcal{B}$ | $a \succ_\$ b =_\$ c$ |  |
| $\mathcal{C}$ | $a =_\$ b \prec_\$ c$ |  |
| $\mathcal{D}$ | $a =_\$ b =_\$ c$ |  |

**Violation 1: Non-Strict Economic Relationships**

With non-strict economic relationships, a node can be both a (direct or indirect) provider and a (direct or indirect) peer to another node. Figure 11 shows a comparison between non-strict and strict economic relationships.

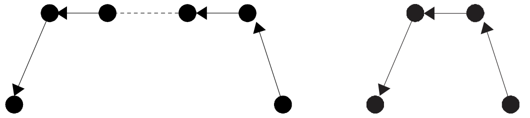


Non-Strict Strict

Fig. 11. Strict and non-strict economic relationships. In the strict variant, node $u$ cannot be an indirect provider and peer to node $z$.

**Violation 2: Transiting Between Peers**

Generally, an AS only carries traffic that is destined to (or originating from) one of its customers. However, due to misconfigurations or complex agreements between peers, an AS may transit traffic between its peers. Economic DPR instances with this violation have an enlarged set of realizable paths. Paths containing valleys of type $\mathcal{D}$ can be adopted by nodes. However, paths are forbidden if they contain valley types $\mathcal{A}$, $\mathcal{B}$, or $\mathcal{C}$. Therefore, every realizable path consists of a series of zero or more ascending customer-to-provider edges, followed by zero or more peer edges, followed by zero or more descending provider-to-customer edges, as shown in Figure 12.

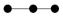
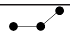
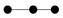
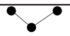
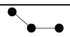


With Peer Transiting Without Peer Transiting

Fig. 12. Allowable paths in economic DPR with and without violation 2.

[1] There are other relaxations that can be considered such as sibling relationships (*i.e.,* backup links) between ASes [10].

TABLE III
VIOLATIONS OF THE ECONOMIC DPR MODEL

| Violation | Valley Types in Causation Chains: | | | | Vertical Cycles | Horizontal Cycles | Potentially Unsafe? |
|---|---|---|---|---|---|---|---|
| | $\mathcal{A}$ | $\mathcal{B}$ | $\mathcal{C}$ | $\mathcal{D}$ | | | |
| 0: None | | | | | simple | none | no |
| 1: Non-Strict Economics | | | | | simple | none | no |
| 2: Transiting | | | |  | simple | non-simple, simple | yes |
| 3: Peers Preferred | | |  |  | simple | non-simple, simple | yes |
| 4: Providers Preferred |  |  | | | non-simple, simple | non-simple, simple | yes |

**Violation 3: Peer Paths over Customer Paths**

Whereas violation 2 is a relaxation on the set of realizable paths, violation 3 is a relaxation of the path preferences. Nodes in economic DPR instances with violation 3 can prefer peer paths over customer paths. Nodes, however, cannot prefer provider paths over peer/customer paths. Only valley-free paths are realizable.

**Violation 4: Provider Paths over Peer/Customer Paths**

Nodes in economic DPR instances with violation 4 can prefer provider paths over peer/customer paths. Again, only valley-free paths are realizable.

### *B. Dynamics Induced by Violations*

The four violations describe different variants of the economic DPR model. Each variant results in different types of causation chains and cycles. For ease of exposition, we model the resulting dynamics of each violation in isolation. The theoretical proofs for violation 2 can be found in Appendix B. The proofs for all the other individual violations, as well as all possible *combinations* of these violations can be found in [9].

Table III summarizes the effects of each violation on the characteristics of causation chains and cycles. The first and second rows show the strict and non-strict economic DPR models. They are the only two variants guaranteed to be safe. The non-strict economic DPR model, however, when combined with other violations could lead to potentially unsafe behavior. The three other violations induce routing behavior which is potentially unsafe.

INTERFERENCEBEAT can be extended using the results of Table III. Upon the detection of a valley in the causation chain, its type ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, or $\mathcal{D}$) can rule out possible causing violations. For example, if a valley of type $\mathcal{B}$ was detected using INTERFERENCEBEAT, then violations 1, 2, and 3 can be immediately ruled out as the possible causes for the observed behavior. Similar methods can be used upon detection of non-simple or horizontal cycles.

## VII. RELATED WORK

Static models for BGP, such as the Stable Paths Problem (SPP) [4], provide insight into the steady-state behavior of policy routing. There are also offline methods that leverage SPP and utilize information from BGP tables [11] to infer policy conflicts between ASes. DPR extends SPP to give insight into the real-time transient behavior of networks. DPR allows us to reason about issues such as misconfigured routing policies or networks with sporadic link failures.

The canonical solution for detecting policy conflicts based on SPP is the Safe Path Vector Protocol (SPVP) introduced by Griffin *et al.* in [12]. SPVP exchanges route flaps among ASes in extended “history” messages that are essentially passed along our causation chains. INTERFERENCEBEAT extends SPVP by appending additional information in a small token to each routing update message to detect violations of the Gao-Rexford guidelines [5].

There are many algorithms that attempt to detect and resolve policy conflicts. Counting [13] and other token-based [14] heuristic approaches benefit from having a low communication overhead. INTERFERENCEBEAT extends such approaches by leveraging the DPR model to guarantee the correctness of policy violation detection and diagnosis.

Finally, there are routing architectures that constrain traditional policy routing to guarantee convergence. Metarouting [15] defines a policy language based on a routing algebra [16] that gives compile-time guarantees of routing convergence. In [17], real-time enforcement of convergence is achieved by passing information in tokens to affect policy rankings. INTERFERENCEBEAT does not enforce convergence. Instead it leverages the DPR model to detect non-compliance to the principles of safe routing dynamics and notifies ASes upon the detection of policy violations.

## VIII. CONCLUSIONS

We introduced a Dynamic Policy Routing (DPR) model, which extends the static model of BGP to capture the *propagation dynamics* of route flaps due to *arbitrary* changes in topology or path preferences. The theoretical results of this paper can be summarized by three key principles which distill the properties of routing dynamics in a safe (economic) policy configuration.

We introduce INTERFERENCEBEAT, a novel distributed algorithm to detect and diagnose policy violations. INTERFERENCEBEAT has a beneficial set of characteristics such as efficiency, privacy, and adoptability. Diagnosis is further enhanced by modelling common policy violations such as the preference of peer paths over customer paths.

## ACKNOWLEDGMENTS

This work has been partially supported by National Science Foundations awards: CISE/CCF #0820138, CISE/CSR #0720604, CISE/CNS #0524477, CNS/ITR #0205294, and CISE/EIA RI #0202067.

## References

[1] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," *IEEE/ACM Trans. Netw.*, vol. 9, pp. 293–306, June 2001.

[2] A. Feldmann, O. Maennel, Z. Mao, A. Berger, and B. Maggs, "Locating Internet Routing Instabilities," in *ACM SIGCOMM*, September 2004.

[3] K. Varadhan, R. Govindan, and D. Estrin, "Persistent Route Oscillations in Inter-domain Routing," *Computer Networks*, vol. 32, pp. 1–16, 2000.

[4] T. Griffin, F. Shepherd, and G. Wilfong, "The Stable Paths Problem and Interdomain Routing," *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 232–243, Apr 2002.

[5] L. Gao and J. Rexford, "Stable Internet Routing Without Global Coordination," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 681–692, 2001.

[6] D. Obradovic, "Real-time Model and Convergence Time of BGP," in *INFOCOM*, 2002.

[7] N. Feamster, H. Balakrishnan, and J. Rexford, "Some Foundational Problems in Interdomain Routing," in *ACM SIGCOMM HotNets*, November 2004.

[8] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*. Cambridge University Press, April 2002.

[9] S. Epstein, K. Mattar, and I. Matta, "Principles of Safe Policy Routing Dynamics," Computer Science Department, Boston University, Tech. Rep. BUCS-TR-2009-13, April 2009.

[10] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, 2001.

[11] F. Wang and L. Gao, "Inferring and Characterizing Internet Routing Policies," in *ACM IMC*, 2003, pp. 15–26.

[12] T. Griffin and G. Wilfong, "A Safe Path Vector Protocol," in *INFOCOM*, 2000, pp. 490–499.

[13] J. Cobb and R. Musunuri, "Enforcing Convergence in Inter-domain Routing," *IEEE GLOBECOM*, vol. 3, pp. 1353–1358, November 2004.

[14] S. Yilmaz and I. Matta, "An Adaptive Management Approach to Resolving Policy Conflicts," in *IFIP Networking*, May 2007.

[15] T. Griffin and J. L. Sobrinho, "Metarouting," in *ACM SIGCOMM*, 2005, pp. 1–12.

[16] J. L. Sobrinho, "An Algebraic Theory of Dynamic Network Routing," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 1160–1173, 2005.

[17] C. Ee and V. Ramachandran and B. Chun and K. Lakshminarayanan and S. Shenker, "Resolving Inter-Domain Policy Disputes," in *ACM SIGCOMM*, 2007, pp. 157–168.

## Appendix A
## Proof of Theorem 1

For convenience of notation, we drop the time index of terms with respect to a given chain $Y = \langle y_0\ y_1\ \dots\ y_k \rangle^t$:

$$\begin{aligned} \pi(y_i) &= \pi(y_i, t+i) \\ \pi_{\text{next}}(y_i) &= \pi(y_i, t+i+1) \\ \rho(y_i) &= \rho(y_i, t+i) \\ \rho_{\text{next}}(y_i) &= \rho(y_i, t+i+1) \\ \text{RankDec}(y_i) &\ \text{iff}\ \ \text{RankDec}(y_i, t+i) \\ \text{RankSame}(y_i) &\ \text{iff}\ \ \text{RankSame}(y_i, t+i) \\ \text{RankInc}(y_i) &\ \text{iff}\ \ \text{RankInc}(y_i, t+i) \end{aligned}$$

**Theorem 1.** Every causation chain of an economic DPR instance $(\succeq_\$, \succeq^t, G)$ is valley-free.

*Proof:* Assume not. Then there exists a causation chain $Y = \langle y_0\ y_1\ \dots\ y_k \rangle^t$ and an index $i$ such that $0 < i < k$ and $y_{i-1} \succeq_\$ y_i \preceq_\$ y_{i+1}$. Thus $y_{i-1}$ and $y_{i+1}$ are peers or providers to $y_i$.

The first part of this proof shows that if this is the case, then at no time during the causation chain did $y_i$ have a customer path. The second part of this proof shows that sometime during the causation chain $y_{i+1}$ had a path through $y_i$. Therefore $y_{i+1}$ had a realized valley path since $y_i$ did not have a customer path and $y_i$ is a customer of or peer to $y_{i+1}$. Since valley-paths are forbidden in economic DPR instances, this results in a contradiction. Since $C(y_i) = y_{i-1}$, either the first or second condition of causation from Table I holds for $y_i$ at time $t + i$.

### Case: $\mathbf{y_i}$ Causation Condition 1

If the first condition of Table I holds for $y_i$ then: $\rho(y_i) = y_{i-1}$ and RankDec$(y_i)$, as shown in Figure 13. Therefore $\pi(y_i) \succ^{t+i} \pi_{\text{next}}(y_i)$. Let $v = \rho_{\text{next}}(y_i)$. It cannot be that $v \prec_\$ y_i$. Otherwise, since $\pi_{\text{next}}(y_i)$ is a customer path and $\pi(y_i)$ is not a customer path (since $\rho(y_i) = y_{i-1} \succeq_\$ y_i$), by the conditions of economic DPR instances: $\pi(y_i) \prec^{t+i} \pi_{\text{next}}(y_i)$, causing a contradiction as shown in Figure 14. Thus $v \succeq_\$ y_i$ and $\rho_{\text{next}}(y_i) \succeq_\$ y_i$.

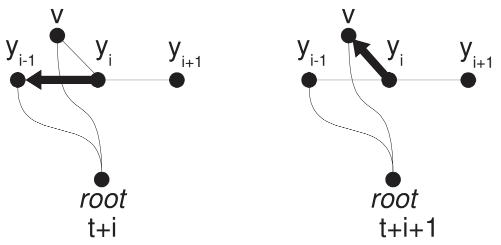


Fig. 13. Causation condition 1: RankDec$(y_i)$

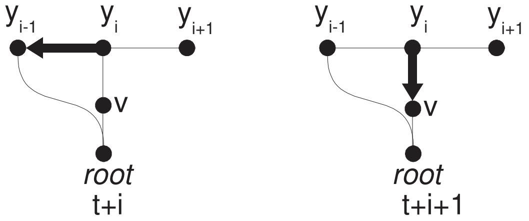


Fig. 14. Contradiction: RankInc$(y_i)$

### Case: $\mathbf{y_i}$ Causation Condition 2

If the second condition of Table I holds for $y_i$ then: $\rho_{\text{next}}(y_i) = y_{i-1}$ and RankInc$(y_i)$, as shown in Figure 15. Therefore $\pi(y_i) \prec^{t+i} \pi_{\text{next}}(y_i)$. Let $v = \rho(y_i)$. It cannot be that $v \prec_\$ y_i$. Otherwise, since $\pi(y_i)$ is a customer path and $\pi_{\text{next}}(y_i)$ is not (since $\rho_{\text{next}}(y_i) = y_{i-1} \succeq_\$ y_i$), by the conditions of economic DPR instances $\pi(y_i) \succ^{t+i} \pi_{\text{next}}(y_i)$, causing a contradiction, as shown in Figure 16. Thus $\rho_{\text{next}}(y_i) \succeq_\$ y_i$ and $v \succeq_\$ y_i$. So for both cases, at no time in the causation chain did $y_i$ have a customer path:

$$\rho(y_i) \succeq_\$ y_i \text{ and } \rho_{\text{next}}(y_i) \succeq_\$ y_i$$

### Case: $\mathbf{y_{i+1}}$ Causation Condition 1

If the first causation condition of Table I holds for $y_{i+1}$, then $\rho(y_{i+1}) = y_i$. By Proposition 1: $\pi(y_{i+1}) =$

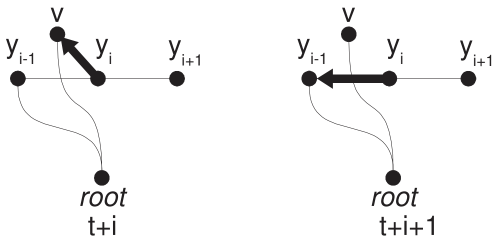


Fig. 15. Causation condition 2: RankInc$(y_i)$

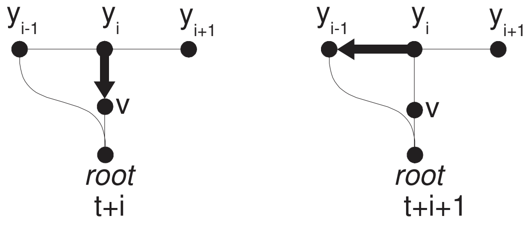


Fig. 16. Contradiction: RankDec$(y_i)$

$\langle y_{i+1}\ \pi(y_i)\rangle$. $\pi(y_{i+1})$ is a valley path since $y_{i+1} \succeq_\$ y_i \preceq_\$ \rho(y_i)$. Since all valley paths are forbidden, $\pi(y_{i+1})$ can never be realized, causing a contradiction.

**Case: $\mathbf{y_{i+1}}$ Causation Condition 2**

Similar arguments can be used if the second causation condition of Table I holds for $y_{i+1}$: $\rho_{\text{next}}(y_{i+1}) = y_i$. Thus by Proposition 1: $\pi_{\text{next}}(y_{i+1}) = \langle y_{i+1}\ \pi_{\text{next}}(y_i)\rangle$. $\pi_{\text{next}}(y_{i+1})$ is a valley path since $y_{i+1} \succeq_\$ y_i \preceq_\$ \rho_{\text{next}}(y_i)$, and can never be realized. Thus in all cases a contradiction occurs, proving the theorem. ■

## Appendix B
## Theorems and Proofs for Violation 2

**Theorem 3.** *Every causation chain in an economic DPR instance with violation 2 does not admit valley types $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$.*

*Proof:* Assume not. Then there exists a causation chain $Y = \langle y_0\ y_1\ \ldots\ y_k\rangle^t$ and an index $i$ such that $0 < i < k$ and at least one of the two conditions hold: (a) $y_{i-1} \succ_\$ y_i \preceq_\$ y_{i+1}$ and/or (b) $y_{i-1} \succeq_\$ y_i \prec_\$ y_{i+1}$.

**Case (a):** $y_{i-1} \succ_\$ y_i \preceq_\$ y_{i+1}$

If case (a) holds, then it can be shown that both $\rho(y_i) \succ_\$ y_i$ and $\rho_{\text{next}}(y_i) \succ_\$ y_i$. This can be seen by looking at the causation conditions of $y_i$. If causation condition 1 holds for $y_i$, then $y_{i-1} = \rho(y_i)$ and RankDec$(y_i)$. It cannot be the case that $\rho_{\text{next}}(y_i) \preceq_\$ y_i$, since this would imply that $y_i$ switched from a provider path through $y_{i-1}$ to a non-provider path, since $y_i \prec_\$ \rho(y_i) = y_{i-1}$ and $y_i \succeq_\$ \rho_{\text{next}}(y_i)$. This would imply RankInc$(y_i)$, causing a contradiction. Thus $\rho(y_i) \succ_\$ y_i$ and $\rho_{\text{next}}(y_i) \succ_\$ y_i$. If causation condition 2 holds for $y_i$, then $y_{i-1} = \rho_{\text{next}}(y_i)$ and RankInc$(y_i)$. It cannot be the case that $\rho(y_i) \preceq_\$ y_i$, since this would imply that $y_i$ switched from a non-provider path to a provider path through $y_{i-1}$, since $y_i \succeq_\$ \rho(y_i)$ and $y_i \prec_\$ \rho_{\text{next}}(y_i) = y_{i-1}$. This would imply RankDec$(y_i)$, causing a contradiction. Thus for both cases, $\rho(y_i) \succ_\$ y_i$ and $\rho_{\text{next}}(y_i) \succ_\$ y_i$.

Thus given the results above, we can prove that $y_{i+1}$ had a realized path with valley type $\mathcal{A}$ or $\mathcal{C}$. If causation condition 1 holds for $y_{i+1}$, then $\pi(y_{i+1}) = \langle y_{i+1}\ \pi(y_i)\rangle$. Since $y_{i+1} \succeq_\$ y_i$ and $y_i \prec_\$ \rho(y_i)$, then $\pi(y_{i+1})$ is a realized path with valley type $\mathcal{A}$ or $\mathcal{C}$, causing a contradiction. If causation condition 2 holds for $y_{i+1}$, then $\pi_{\text{next}}(y_{i+1}) = \langle y_{i+1}\ \pi_{\text{next}}(y_i)\rangle$. Since $y_{i+1} \succeq_\$ y_i$ and $y_i \prec_\$ \rho_{\text{next}}(y_i)$, then $\pi_{\text{next}}(y_{i+1})$ is a realized path with valley type $\mathcal{A}$ or $\mathcal{C}$, causing a contradiction.

**Case (b):** $y_{i-1} \succeq_\$ y_i \prec_\$ y_{i+1}$

If case (b) holds, then using an argument similar to case (a) it can be shown that both $\rho(y_i) \succeq_\$ y_i$ and $\rho_{\text{next}}(y_i) \succeq_\$ y_i$. We can then prove that $y_{i+1}$ had a realized path with valley type $\mathcal{A}$ or $\mathcal{B}$, causing a contradiction. ■

**Theorem 4.** *Every vertical causation cycle $Y = \langle y_0\ \ldots\ y_k\rangle^t$ in an economic DPR instance with violation 2 is simple.*

*Proof:* This proof proceeds by determining $y_1$'s economic relationship with $y_0$ and $y_{k-1}$'s economic relationship with $y_k = y_0$. Since $Y$ is a vertical causation cycle, there exists a minimal index $i$, $0 < i < k$ such that $y_i \neq_\$ y_{i-1}$. Note that $i \neq k$, otherwise $y_0 =_\$ y_1 =_\$ \ldots =_\$ y_{k-1} \neq_\$ y_k$, implying $y_0 \neq_\$ y_k$, which is a contradiction. Either $y_i \succ_\$ y_{i-1}$ or $y_i \prec_\$ y_{i-1}$. It cannot be that $y_{i-1} \succ_\$ y_i$, since by Theorem 3 $y_0 =_\$ y_{i-1} \succ_\$ y_i \succ_\$ y_{i+1} \ldots \succ_\$ y_k$, implying $y_0 \succ_\$ y_k$ which is a contradiction. Therefore $y_{i-1} \prec_\$ y_i$. If $i > 1$, then $y_{i-2} =_\$ y_{i-1} \prec_\$ y_i$, representing a valley of type $\mathcal{C}$, which is a contradiction. So $i = 1$ and $y_0 \prec_\$ y_1$.

Let $j$ be the first index $1 < j < k$ where $y_{j-1} \succ_\$ y_j$. Note that $j$ has to exist otherwise $y_0 \prec_\$ y_1 \preceq_\$ \ldots \preceq_\$ y_k$, implying $y_0 \prec_\$ y_k$ which is a contradiction. From Theorem 3, $y_{h-1} \succ_\$ y_h$ for all $j < h \leq k$. So $y_{k-1} \succ_\$ y_k = y_0$. Therefore $Y$ must be simple, otherwise $\langle y_{k-1}\ y_0\ y_1\rangle$ must be a causation chain. However since $y_{k-1} \succ_\$ y_0$ and $y_0 \prec_\$ y_1$, $Y$ contains a valley of type $\mathcal{A}$, contradicting Theorem 3, and thus proving the theorem. ■

**Theorem 5.** *An economic DPR instance with violation 2 admits simple, non-simple horizontal causation cycles, and is potentially unsafe.*

*Proof:* From the example in Figure 17 representing "Bad Gadget" that is known to have no stable assignment [4]. ■

Path preferences:

| | |
|---|---|
| Node $a$: | $\langle a\ b\ root\rangle$ |
| | $\langle a\ root\rangle$ |
| Node $b$: | $\langle b\ c\ root\rangle$ |
| | $\langle b\ root\rangle$ |
| Node $c$: | $\langle c\ a\ root\rangle$ |
| | $\langle c\ root\rangle$ |

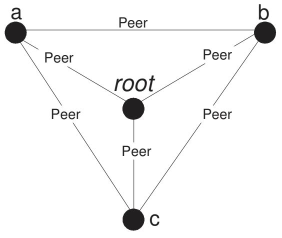


Fig. 17. An economic DPR instance with violation 2.