

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

L. Cittadini, G. Di Battista,
M. Rimondini, S. Vissicchio
Roma Tre University

Background: Routing Policies

- What does a routing policy look like?
- It boils down to
 - route filtering
 - route ranking
- A simple BGP policy might look like this

```
ip as-path access-list 1 deny _31337_
```

```
ip as-path access-list 1 permit .*$
```

```
route-map myPolicy permit 10
```

```
match as-path 1
```

```
set local-preference 120
```

route
Filtering

route
Ranking

Autonomy and Expressiveness

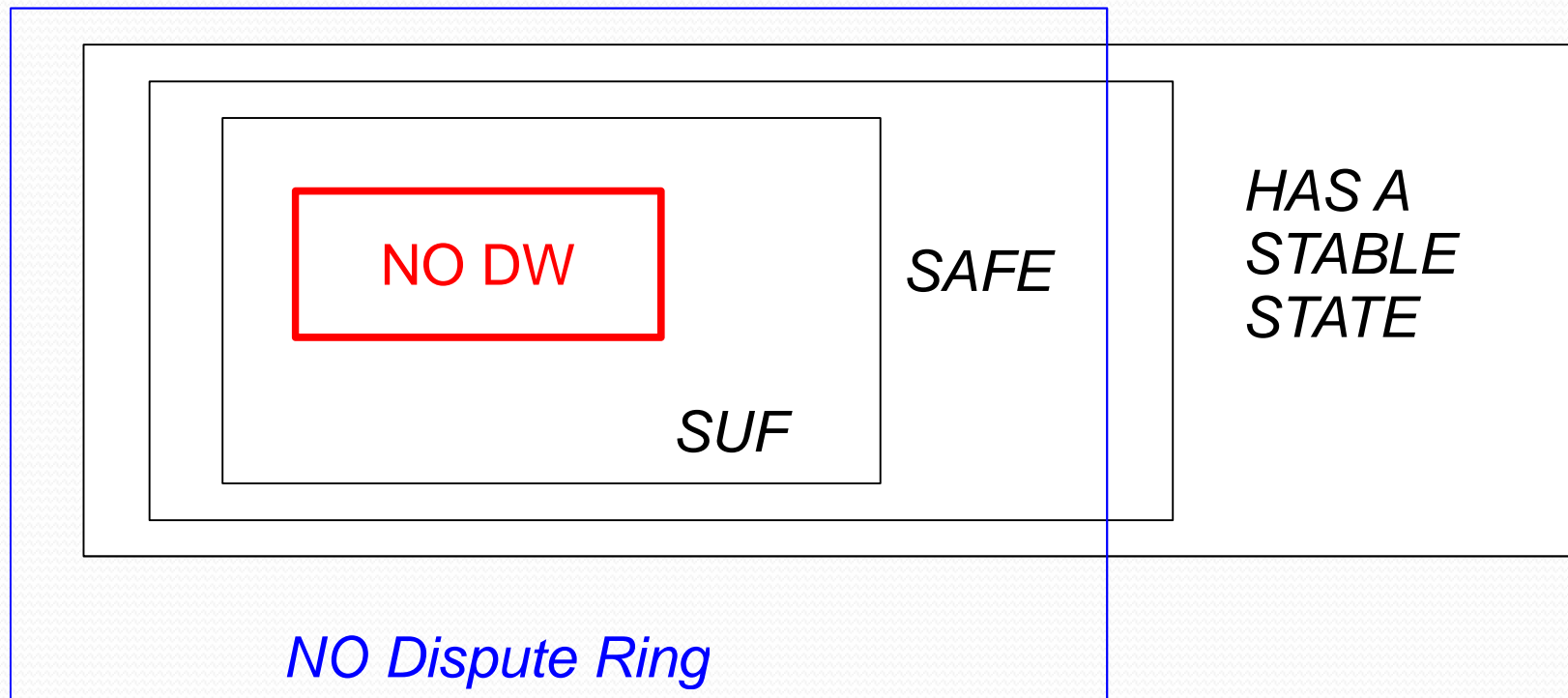
- BGP provides the **expressiveness** to specify arbitrary rankings and filters
- ASes can **autonomously** specify their routing policies
- Consequence: BGP policies **can** [GriffinShepherdWilfong02] and **do** [Carlzon06, Berger01] lead to oscillations

- Problem
 - assume that ASes preserve complete autonomy and filtering expressiveness
 - how expressive can rankings be?
- AKA Safety Under Filtering (SUF) problem [FeamsterJohariBalakrishnan05]

Definitions and State of the Art

- Safety
 - a BGP configuration is safe if it is free from oscillations
 - a **sufficient** condition for safety (“no dispute wheel”) is given in [GriffinShepherdWilfong99,02]
- Safety under Filtering (SUF)
 - a BGP configuration is SUF if it is safe under any combination of routing filters
 - a **necessary** condition for SUF (“no dispute ring”) is given in [FeamsterJohariBalakrishnan05]

The “big picture”

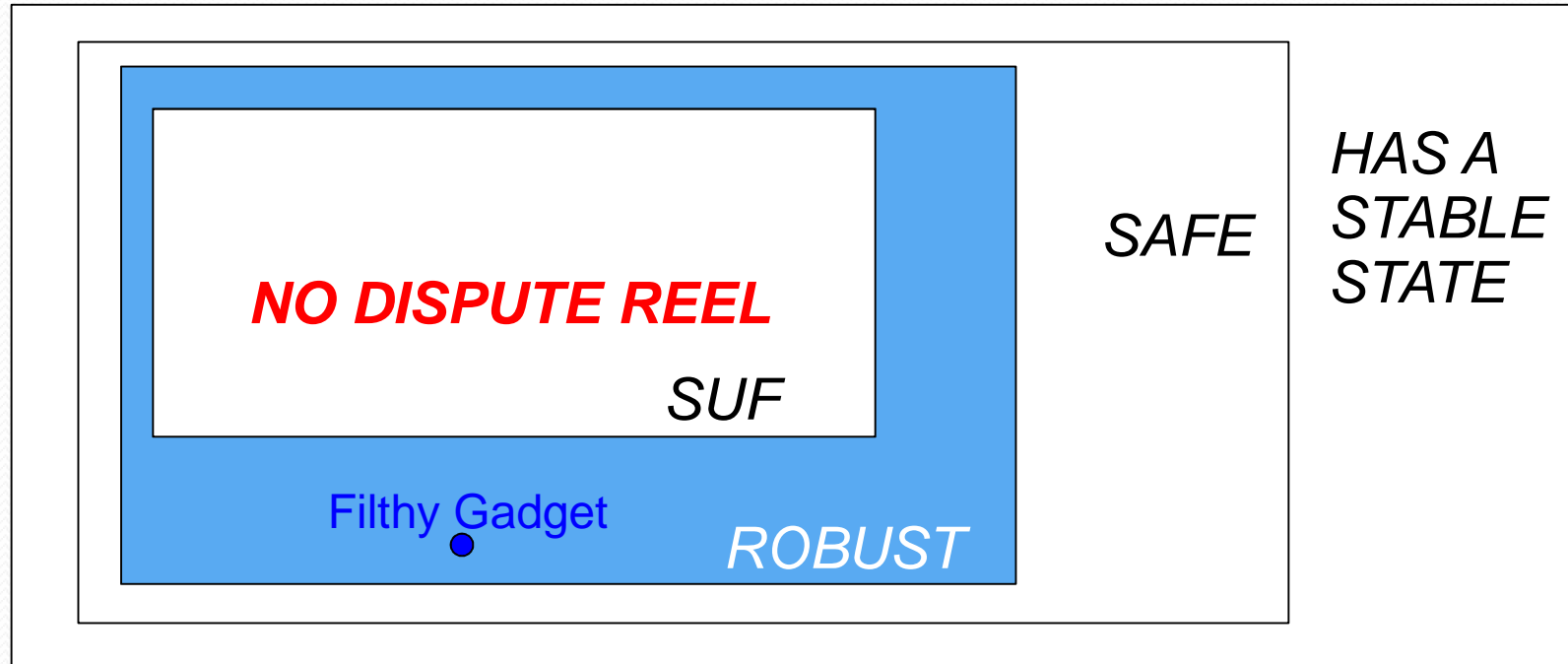


Motivation: Fill the large gap between known necessary and sufficient conditions for stability

Our Contributions

- a characterization for SUF
 - we bind SUF to a particular structure (Dispute Reel) which depends only on the policies and the network topology
 - does not depend on protocol dynamics
- a proof that robustness does not imply SUF
 - robustness is safety under any link failures
 - filters are more dangerous than link failures!
- a debugging technique for multiple stable states
 - multiple stable states imply unsafety [SamiSchapiraZohar09]
 - given two stable states, it is possible find a DR
 - i.e., pinpoint the trouble points in the configuration

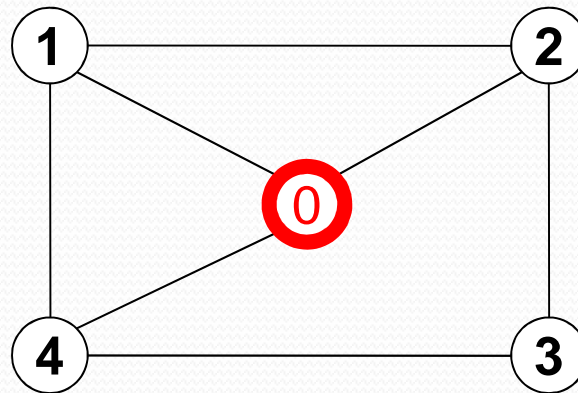
The new “big picture”



Deeper understanding of the interplay between autonomy, expressiveness, and stability

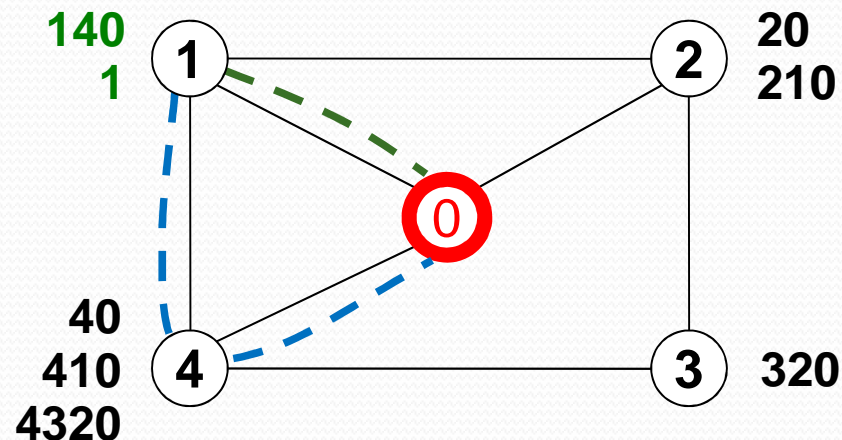
The Model: SPP

- BGP peerings are represented by a graph
 - the destination prefix is originated by special node **0**



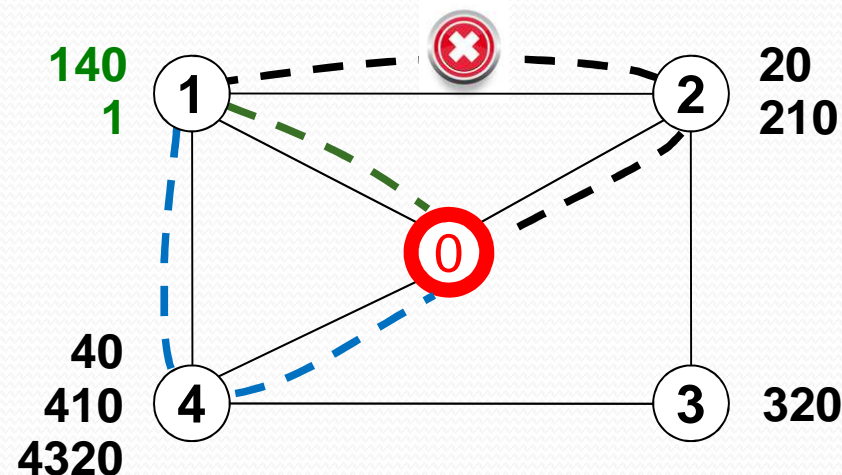
The Model: SPP

- BGP peerings are represented by a graph
 - the destination prefix is originated by special node 0
- BGP policies are encoded in lists of paths



The Model: SPP

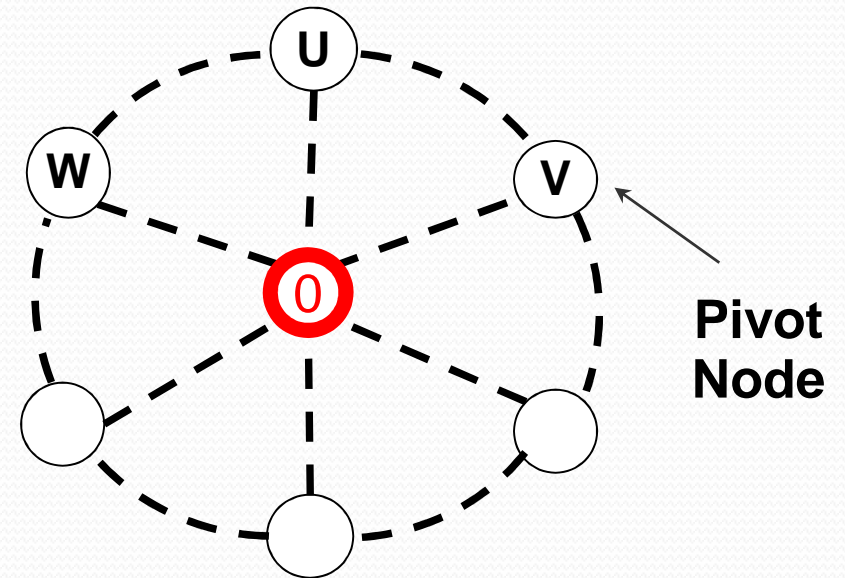
- BGP peerings are represented by a graph
 - the destination prefix is originated by special node 0
- BGP policies are encoded in lists of paths
 - paths that do not appear in a list have been filtered



Wheels



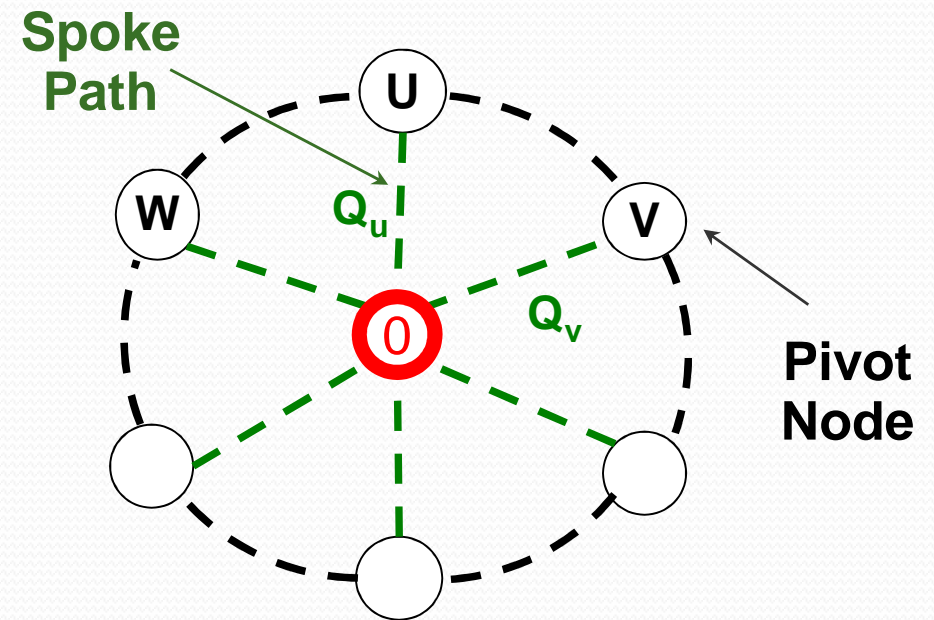
- A Dispute Wheel is a cyclic structure of preferences
- the structure is made of pivot nodes



Wheels



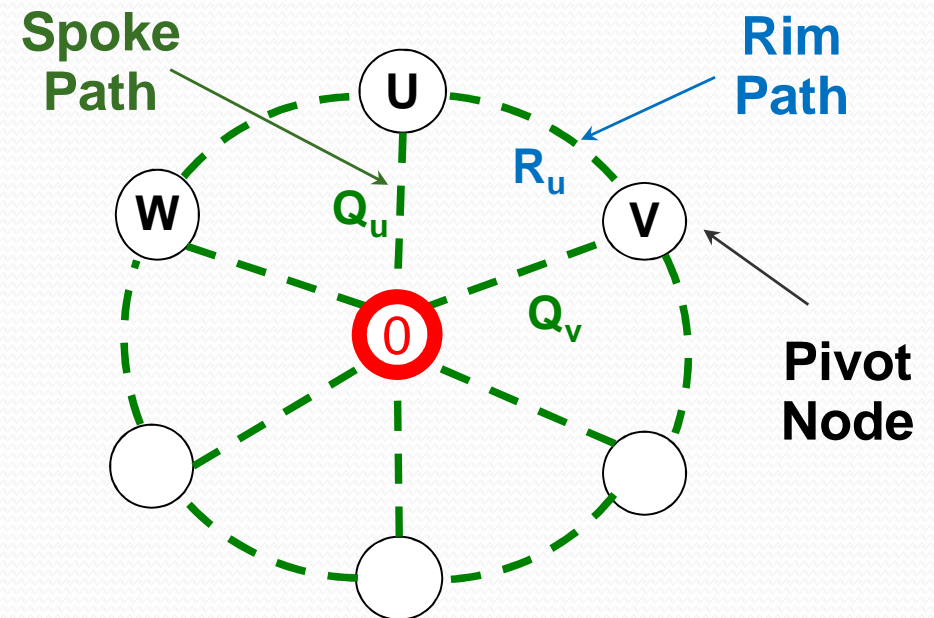
- A Dispute Wheel is a cyclic structure of preferences
- the structure is made of pivot nodes
- each pivot has a direct route



Wheels



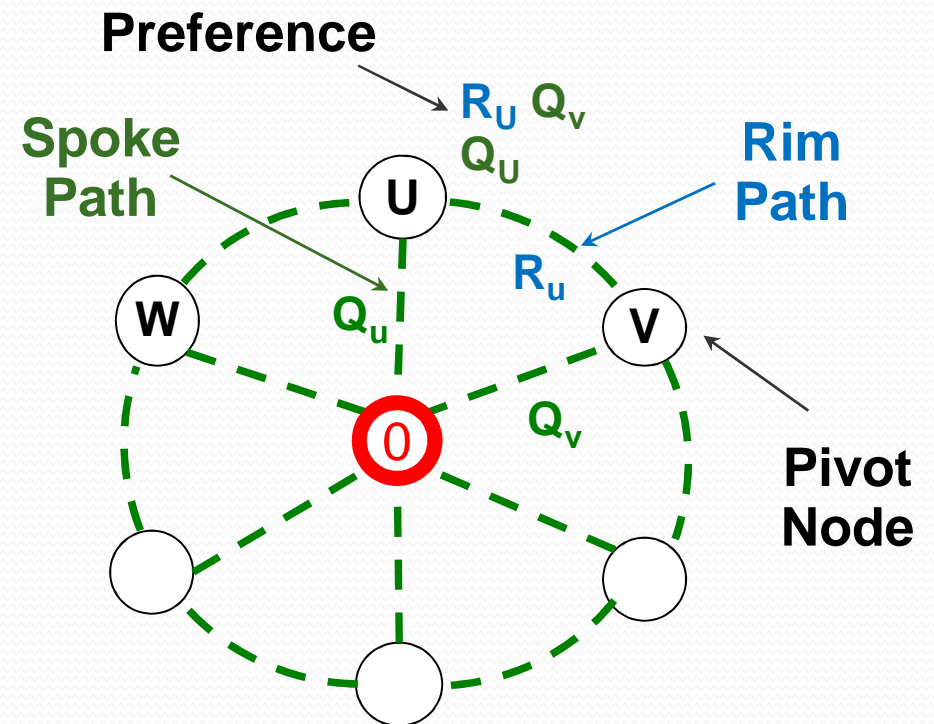
- A Dispute Wheel is a cyclic structure of preferences
 - the structure is made of pivot nodes
 - each pivot has a direct route
 - each pivot has a route via its successor



Wheels



- A Dispute Wheel is a cyclic structure of preferences
 - the structure is made of pivot nodes
 - each pivot has a direct route
 - each pivot has a route via its successor
 - each pivot prefers the route via its successor to the direct route

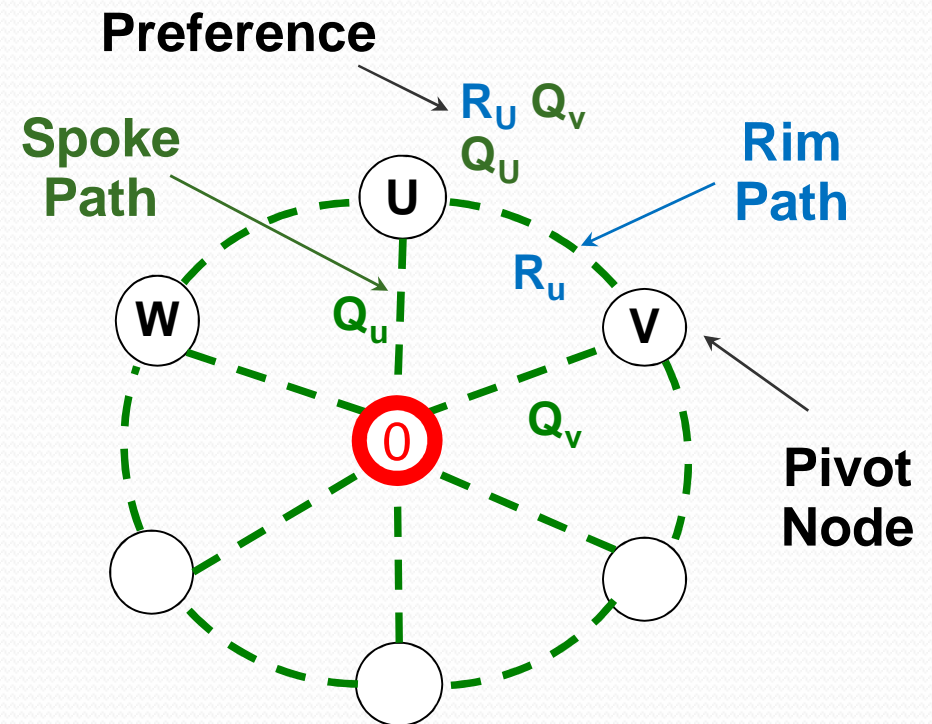


Wheels



- A Dispute Wheel is a cyclic structure of preferences

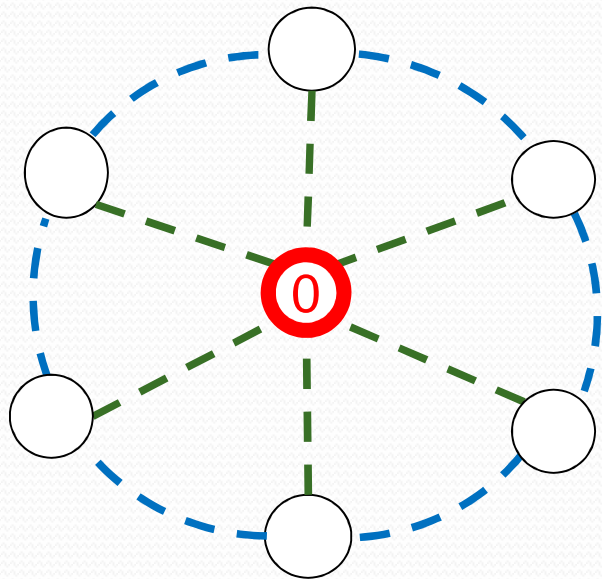
- the structure is made of pivot nodes
- each pivot has a direct route
- each pivot has a route via its successor
- each pivot prefers the route via its successor to the direct route



- No Dispute Wheel \Rightarrow SUF [GriffinShepherdWilfong99,02]

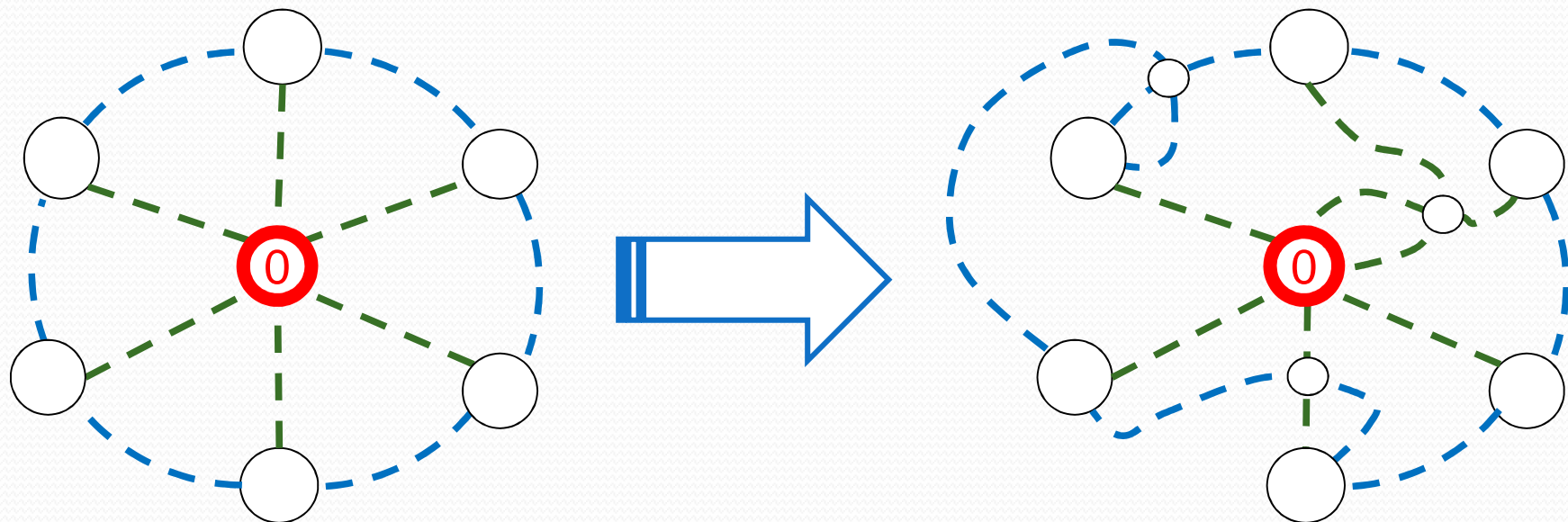
Ideal vs Real Wheels

- When we think about a DW, we imagine it to be clean and simple



Ideal vs Real Wheels

- When we think about a DW, we imagine it to be clean and simple
- Actually, a DW in a BGP configuration could be quite complex
 - e.g., intersections between spoke/rim path



Rings

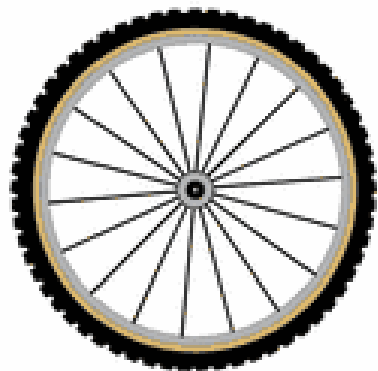


- A Dispute Ring is a DW such that each node appears only once in the wheel
- $SUF \Rightarrow$ No Dispute Ring
[FeamsterJohariBalakrishnan05]
- Intuition
 - meet in the middle to characterize SUF

Rings



- A Dispute Ring is a DW such that each node appears only once in the wheel
- $SUF \Rightarrow$ No Dispute Ring
[FeamsterJohariBalakrishnan05]
- Intuition
 - meet in the middle to characterize SUF



too complex

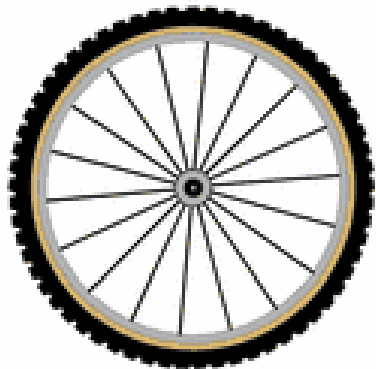


too simple

Rings



- A Dispute Ring is a DW such that each node appears only once in the wheel
- $SUF \Rightarrow$ No Dispute Ring
[FeamsterJohariBalakrishnan05]
- Intuition
 - meet in the middle to characterize SUF



too complex



right!



too simple

Wheel + Ring = Reel

- A Dispute Reel (DR) is a particular kind of DW and a generalization of a Dispute Ring.
- A DR is a DW such that
 1. Pivot vertices appear in exactly three paths
 2. **Spoke** and **rim** paths do not intersect
 3. **Spoke** paths form a tree
 - only intersections among **rim** paths are allowed
- A DW that does not satisfy these conditions does not pose stability problems

Roadmap to characterization

- Theorem (5.1): No DR \Rightarrow SUF
 - (not shown in this presentation)
 - an oscillation implies a DW having special properties [GriffinShepherdWilfong02]
 - such a DW is (or contains) a DR

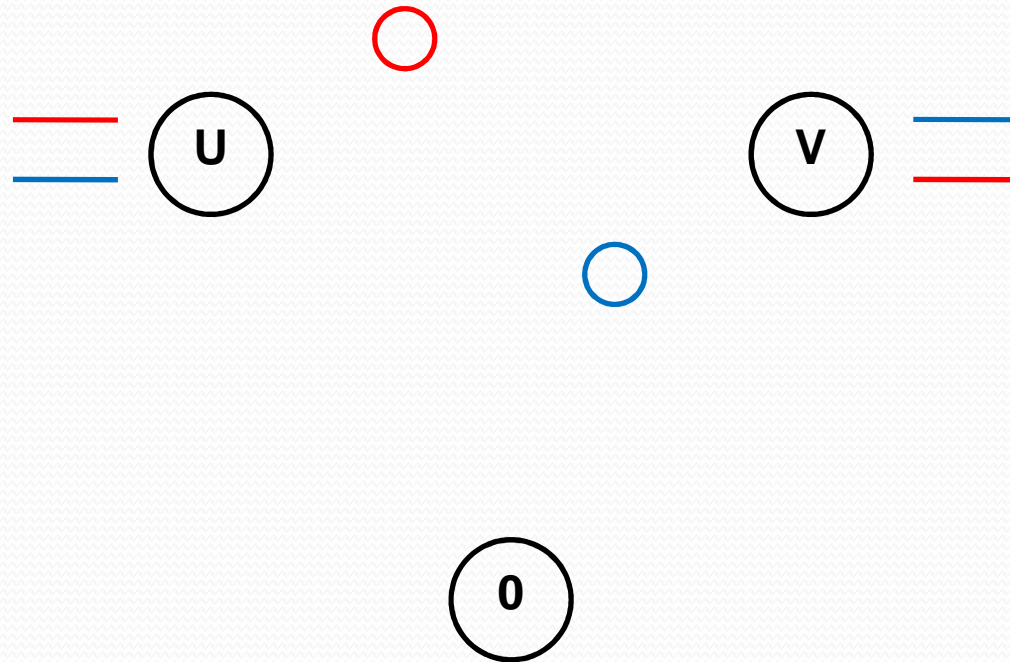
- Theorem (4.1): Existence of a DR \Rightarrow Not SUF
 - we first apply filters to the network
 - then we show a routing oscillation

DR => Not SUF

- The proof is based on two different “types” of routing oscillations
 - synchronous oscillations for DR having 2 pivots
 - asynchronous oscillations for DR with >2 pivots

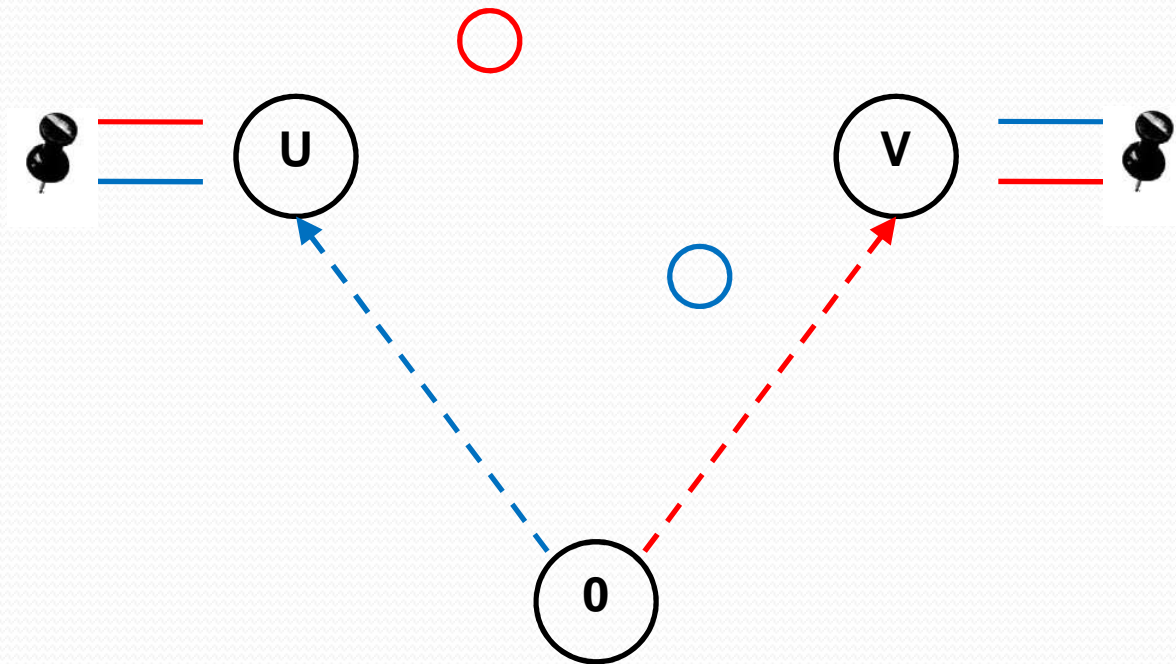
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



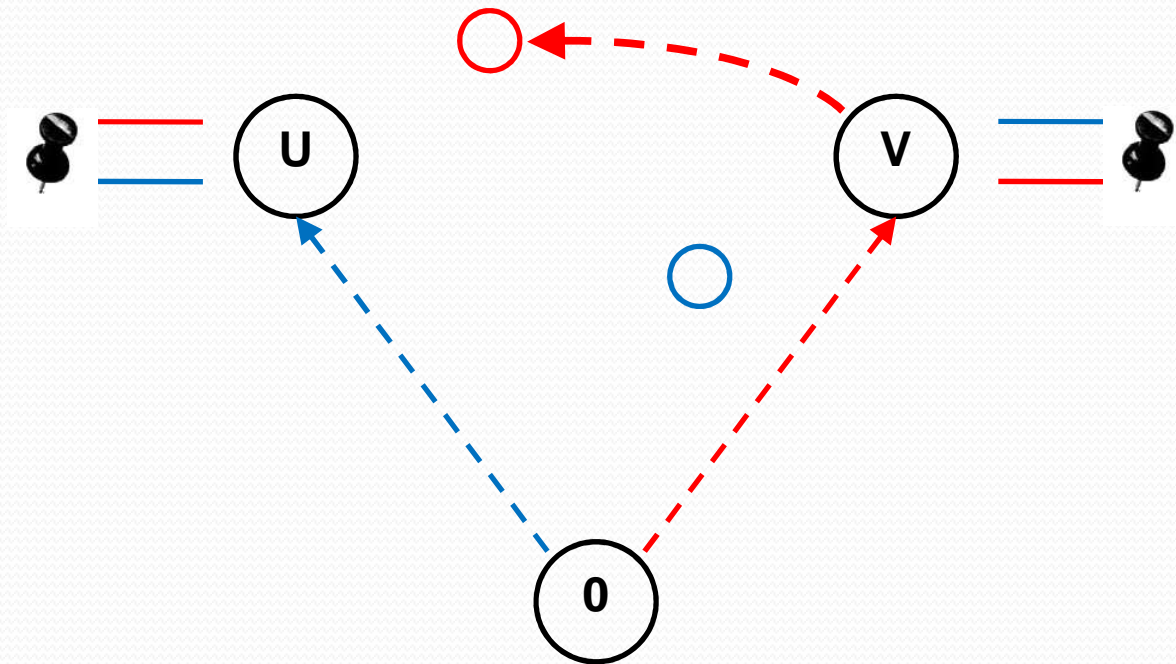
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



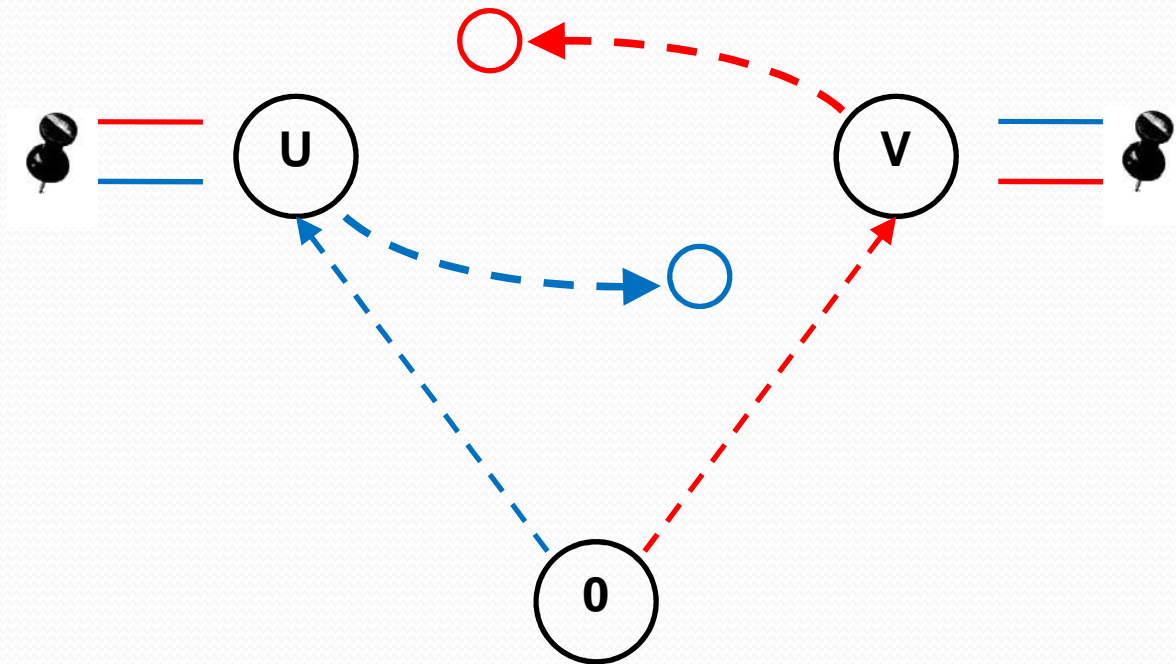
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



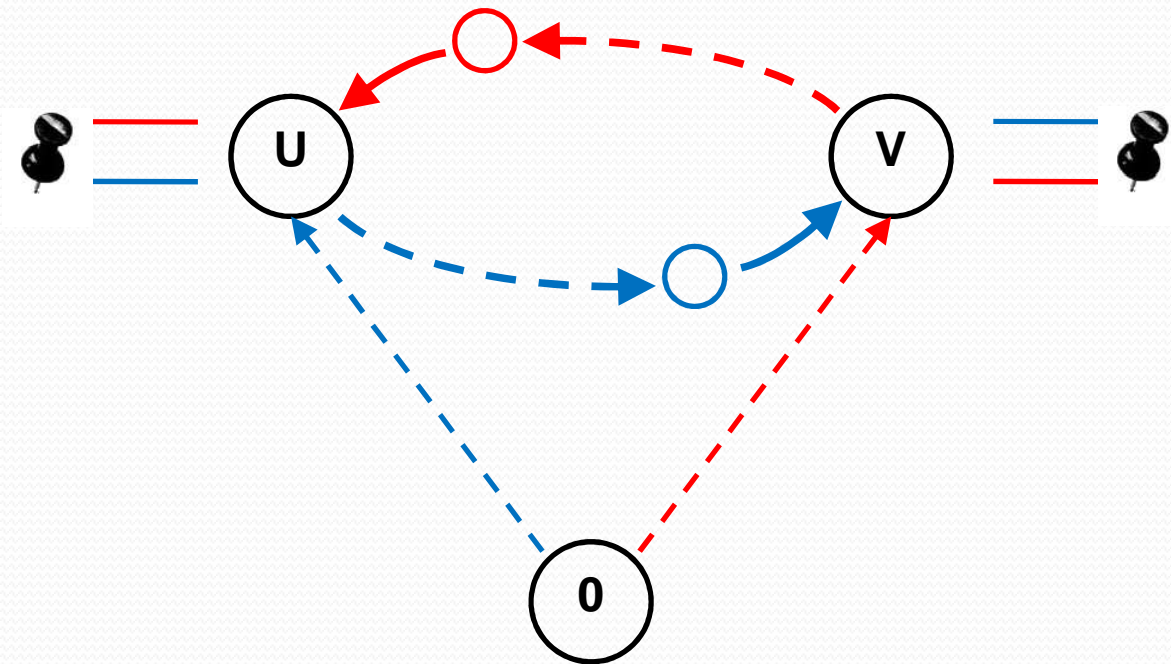
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



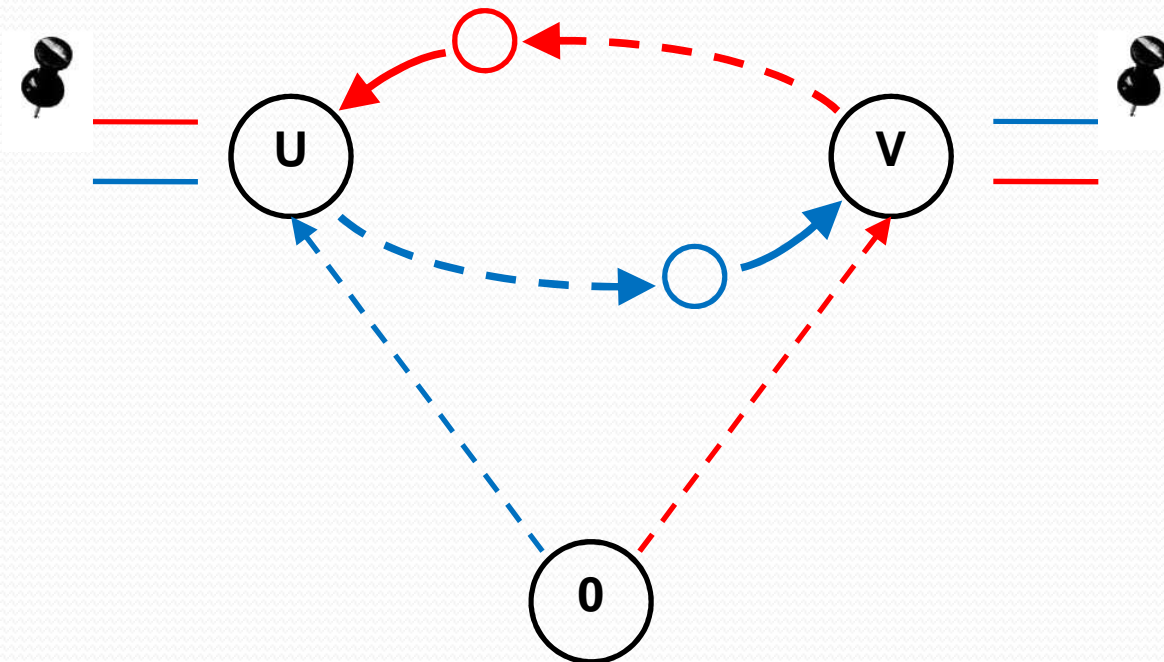
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



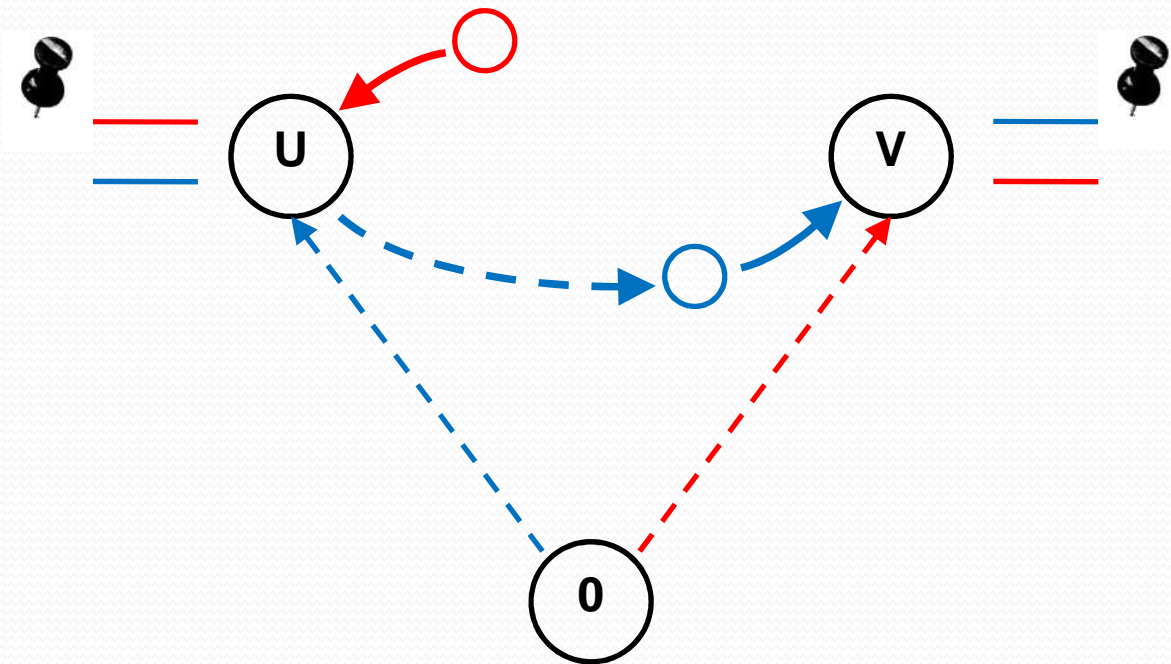
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



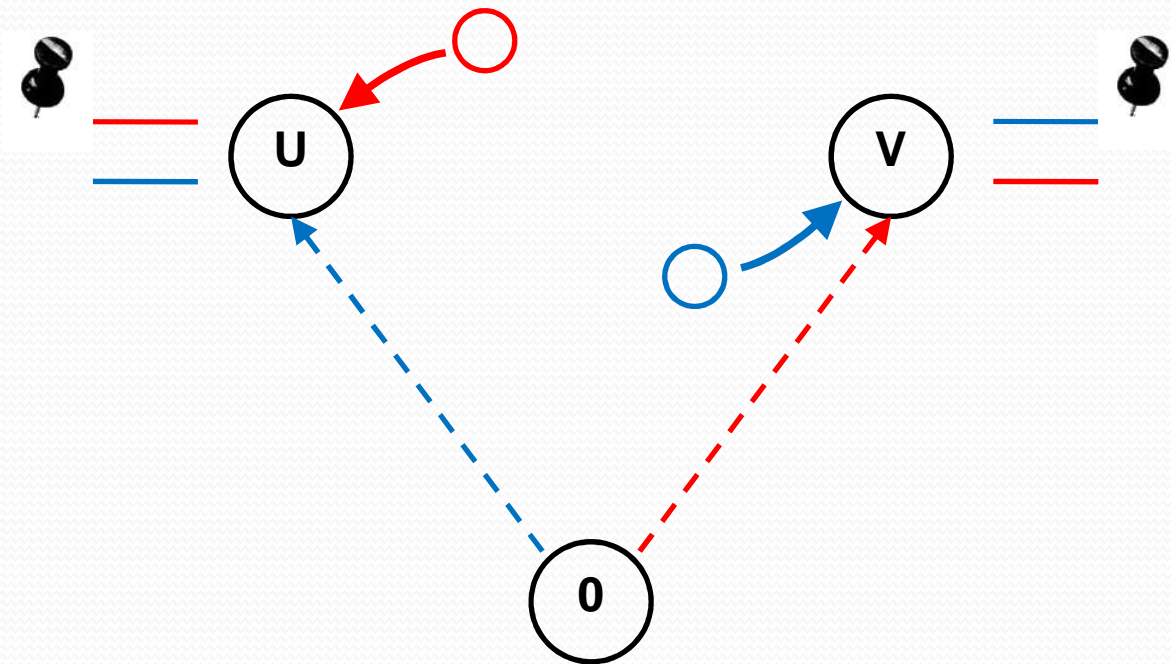
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



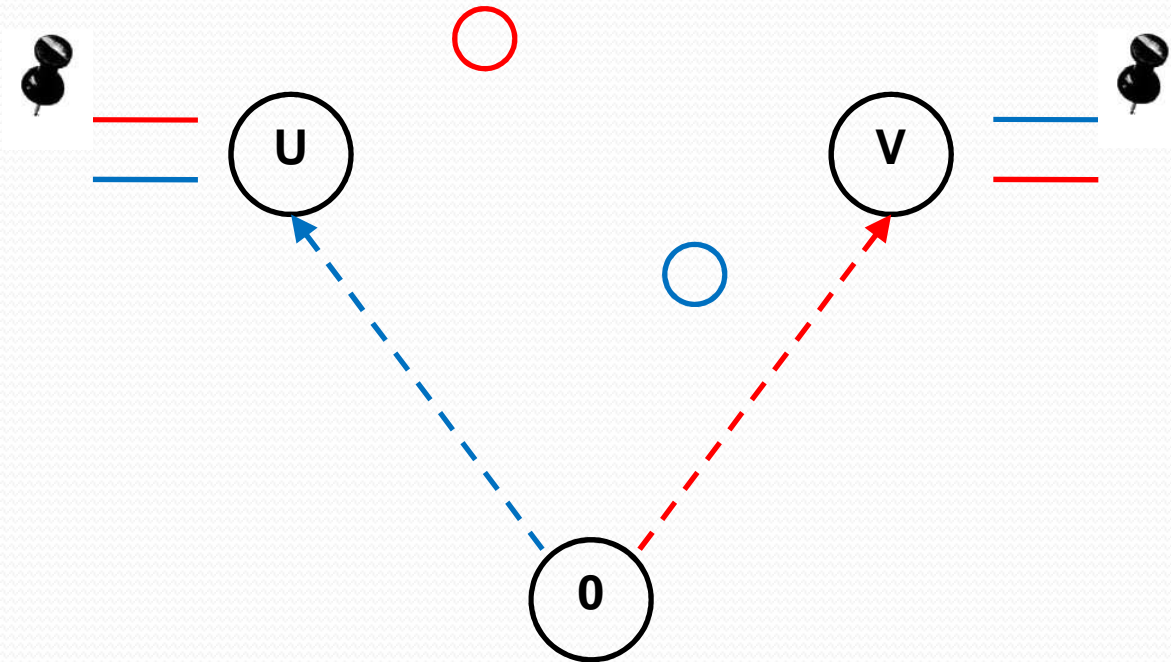
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



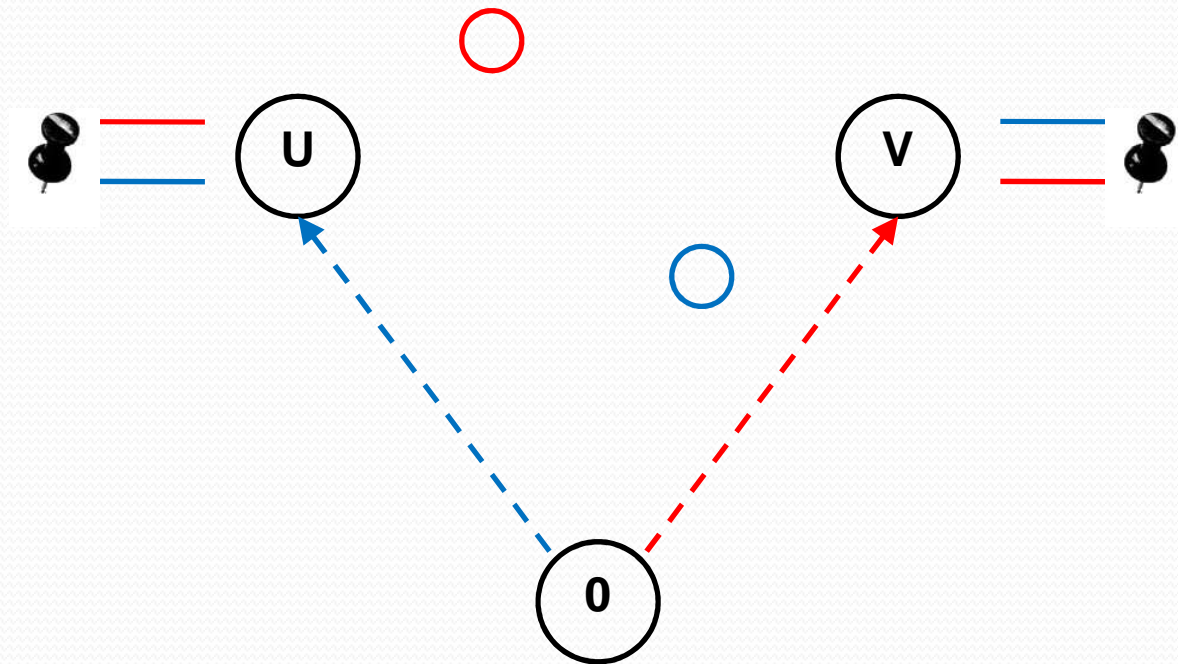
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



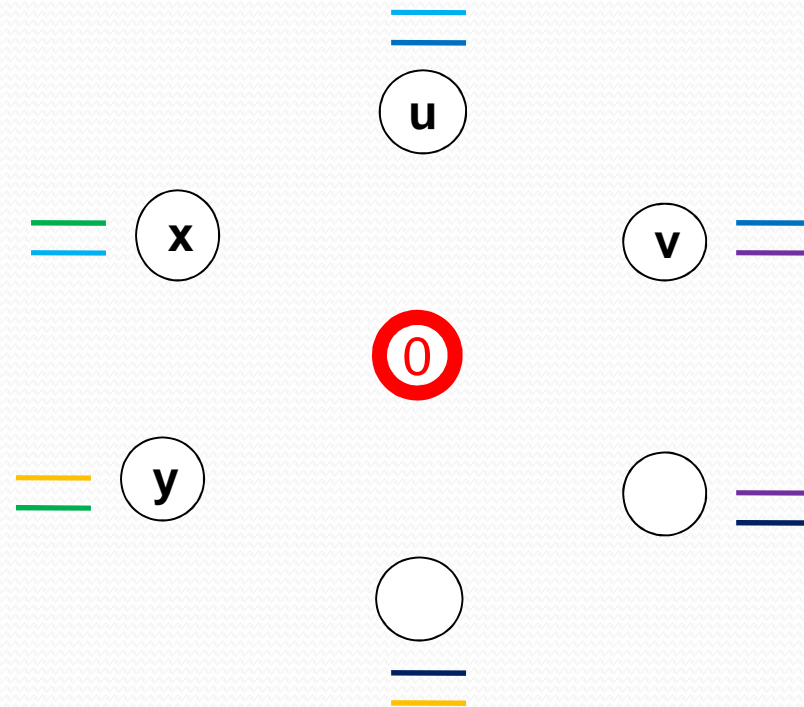
DRs with only 2 pivots

- Lemma (4.1): A DR consisting of two pivots can oscillate
 - filter any path that does not appear in the DR
 - synchronous oscillation



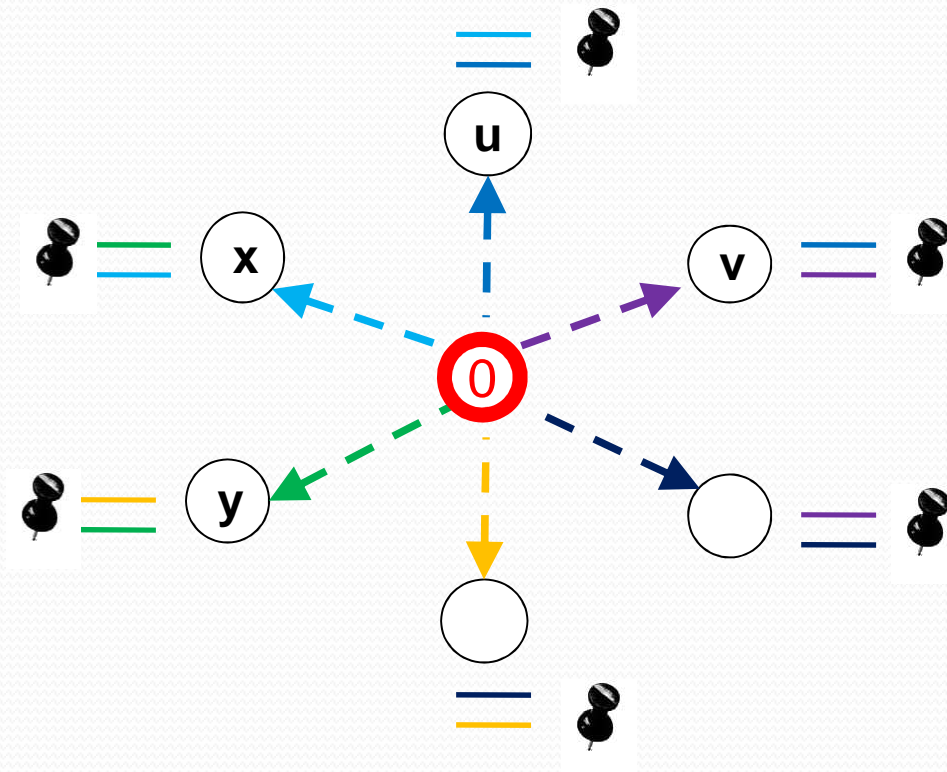
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



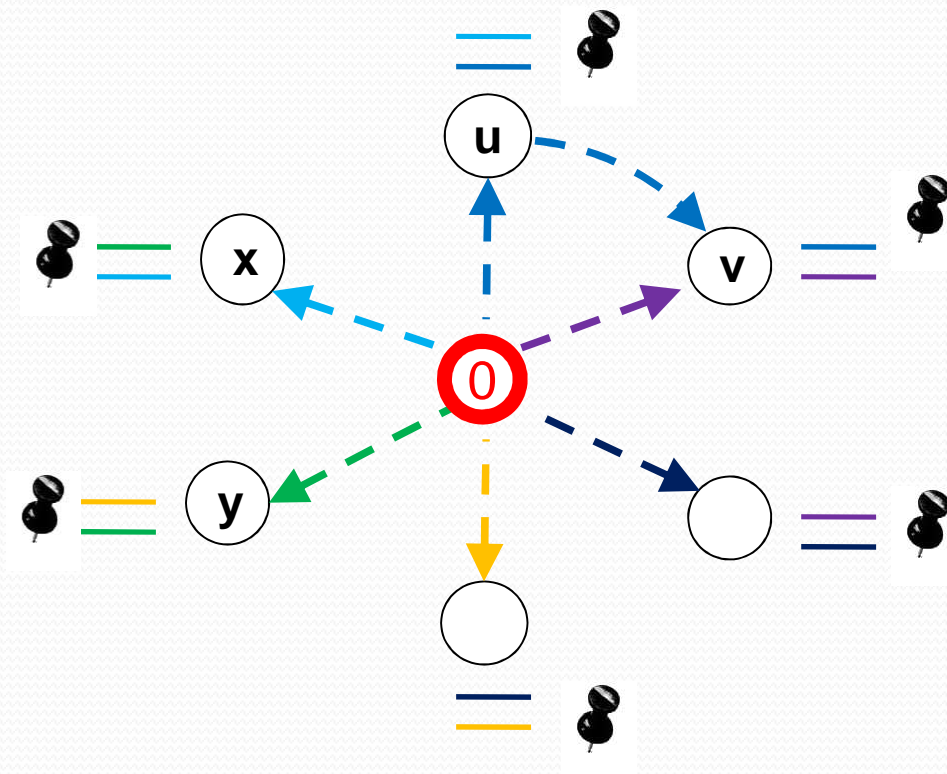
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



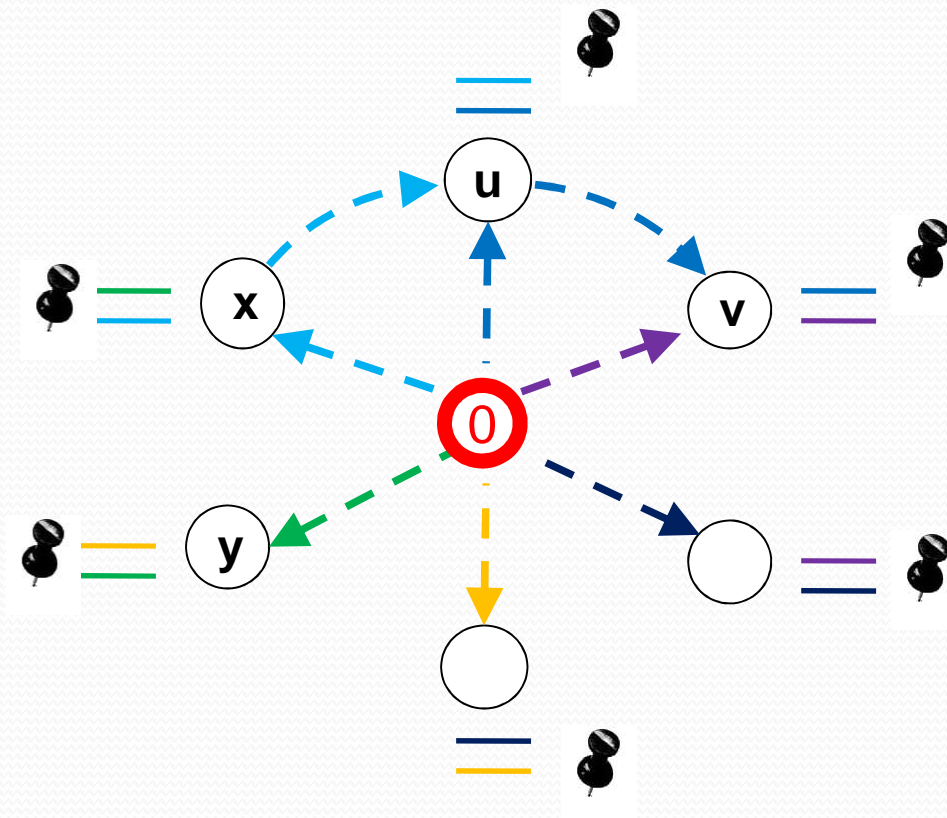
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



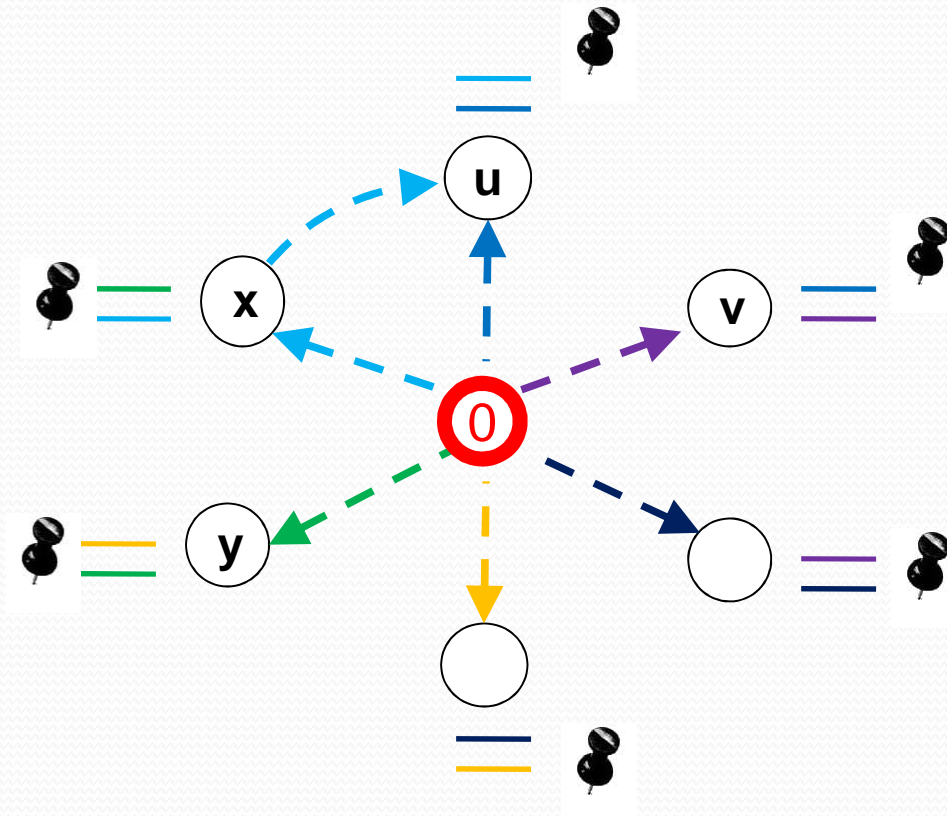
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



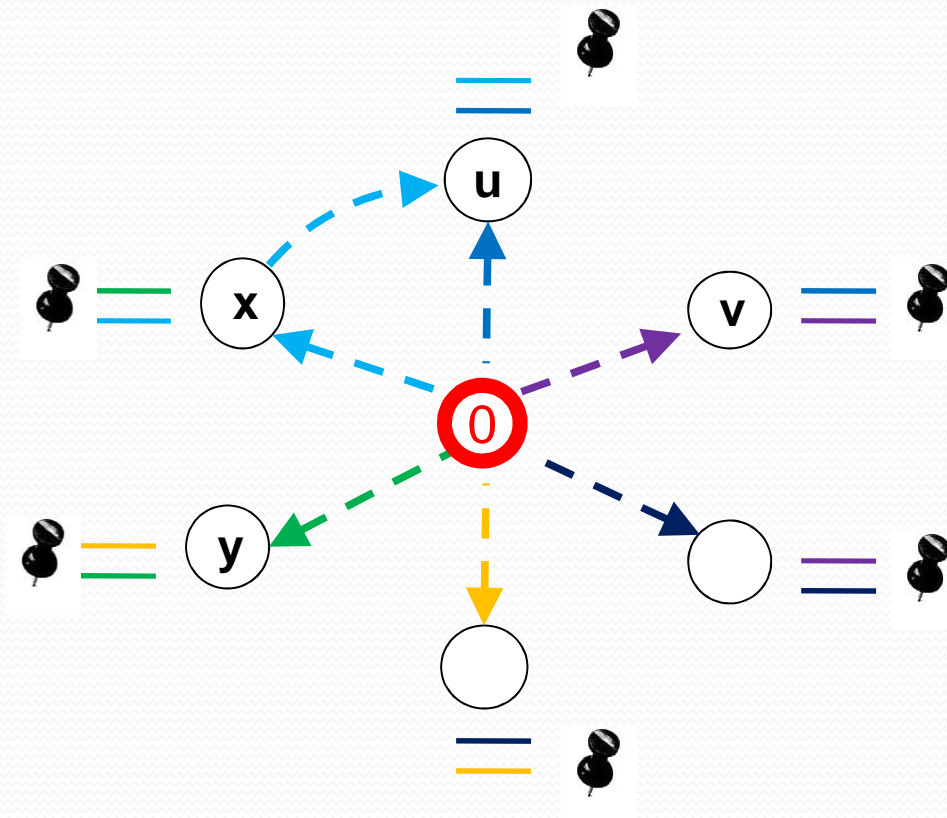
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



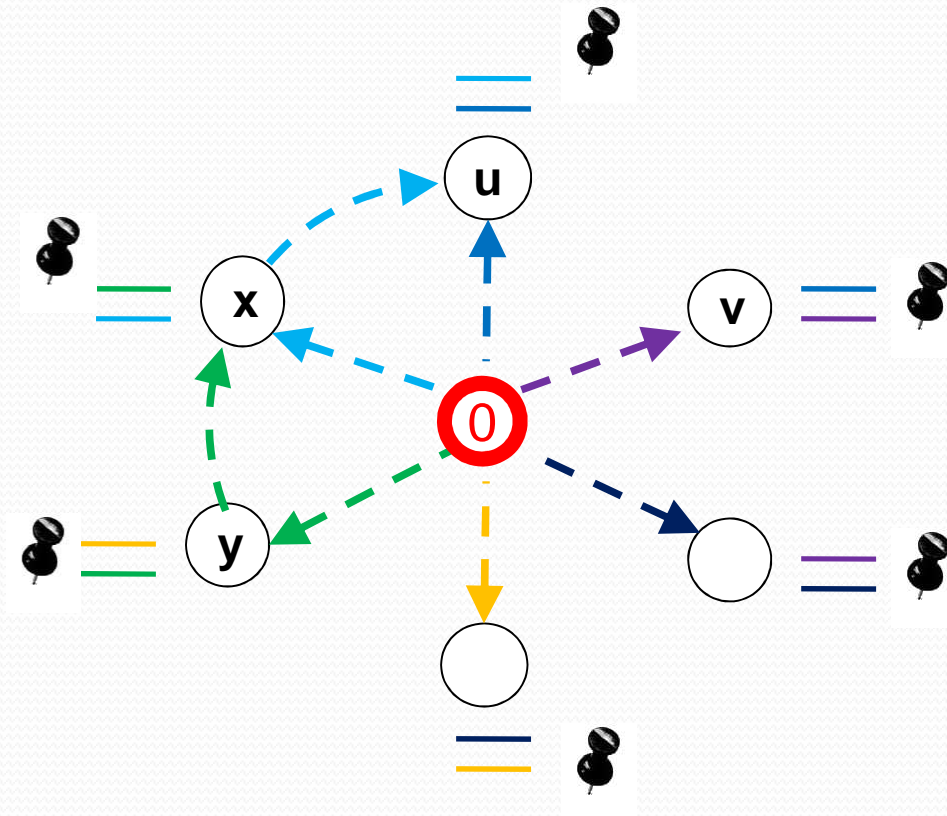
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



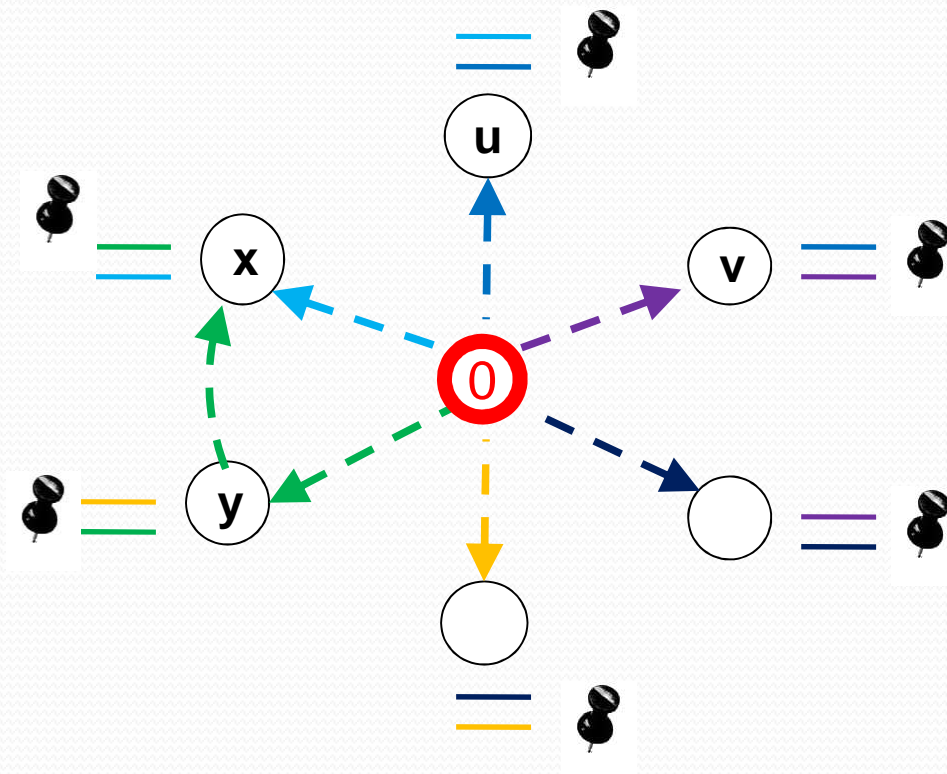
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



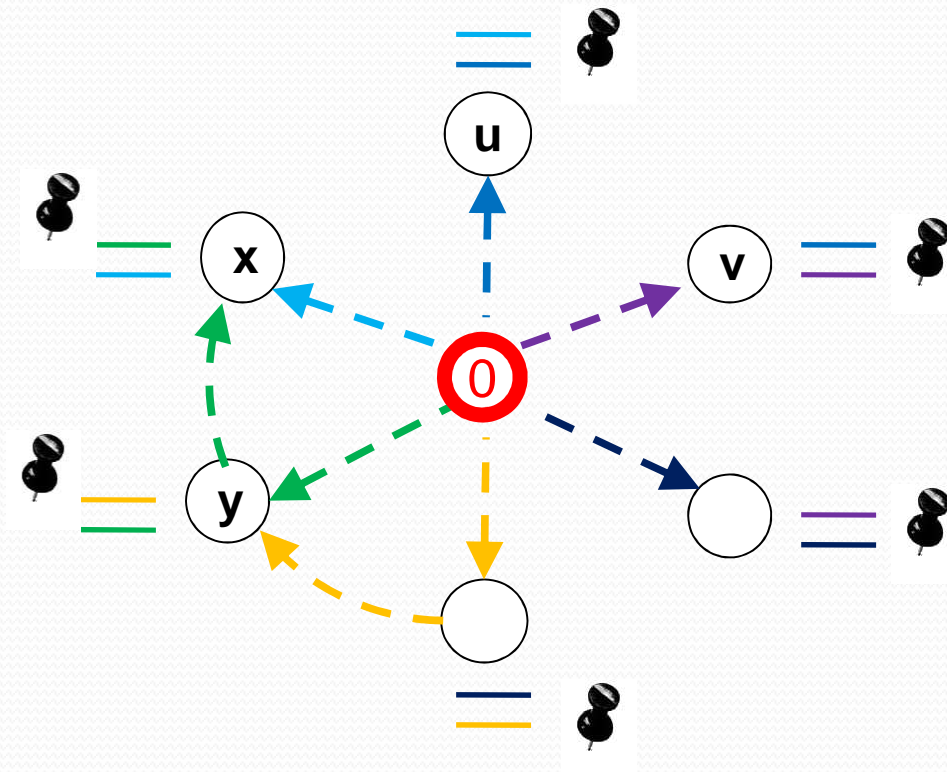
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



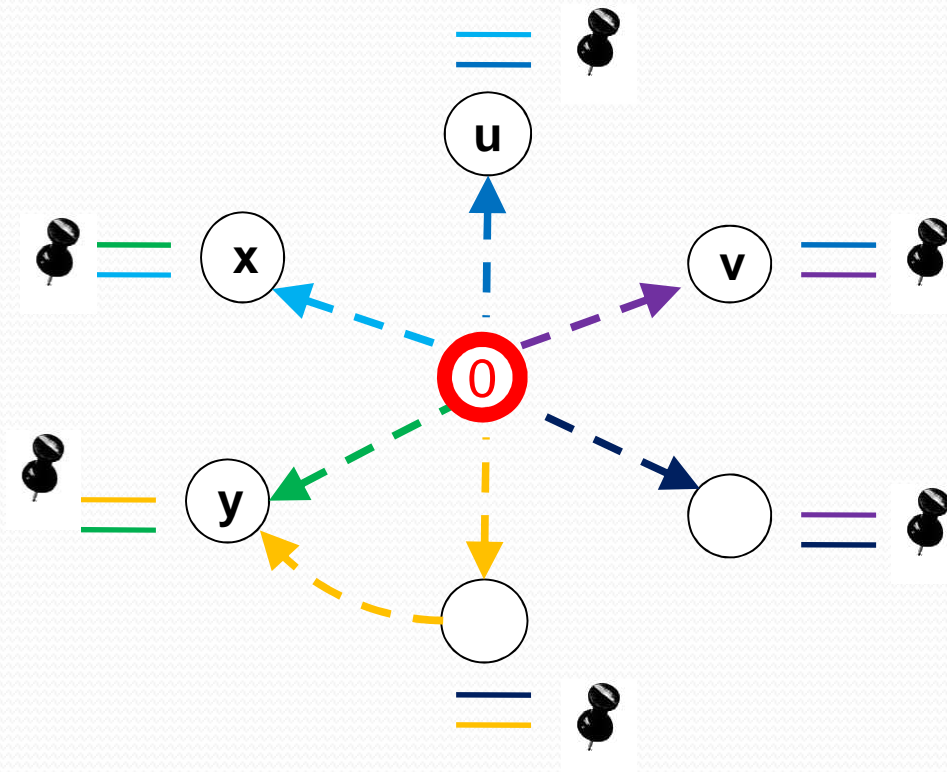
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



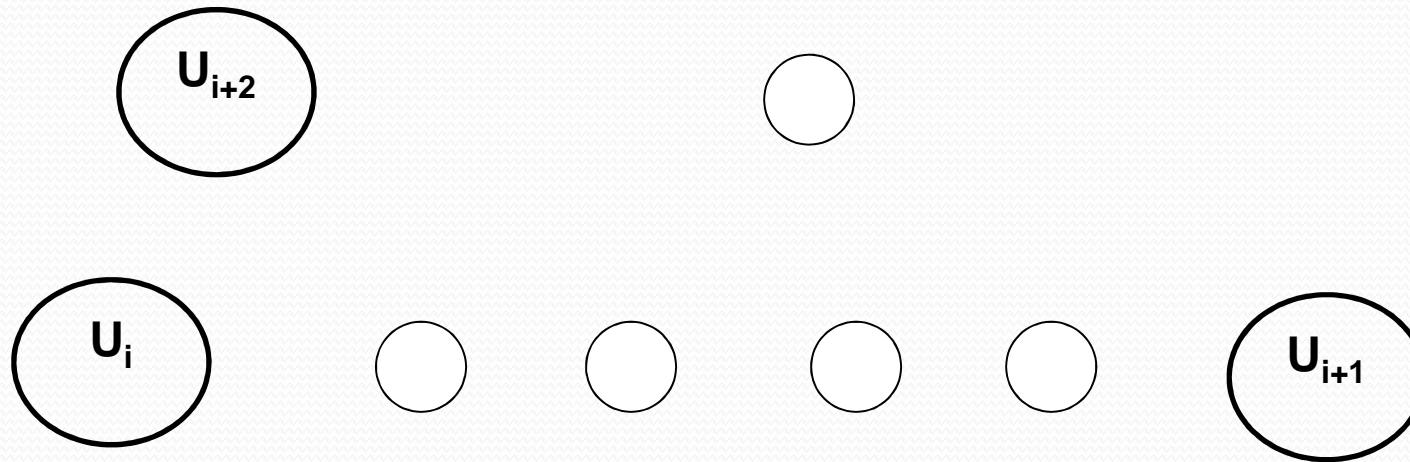
DRs with >2 pivots

- Lemma (4.3): A DR with >2 pivots can oscillate
 - filter any path that does not appear in the DR
 - asynchronous oscillation



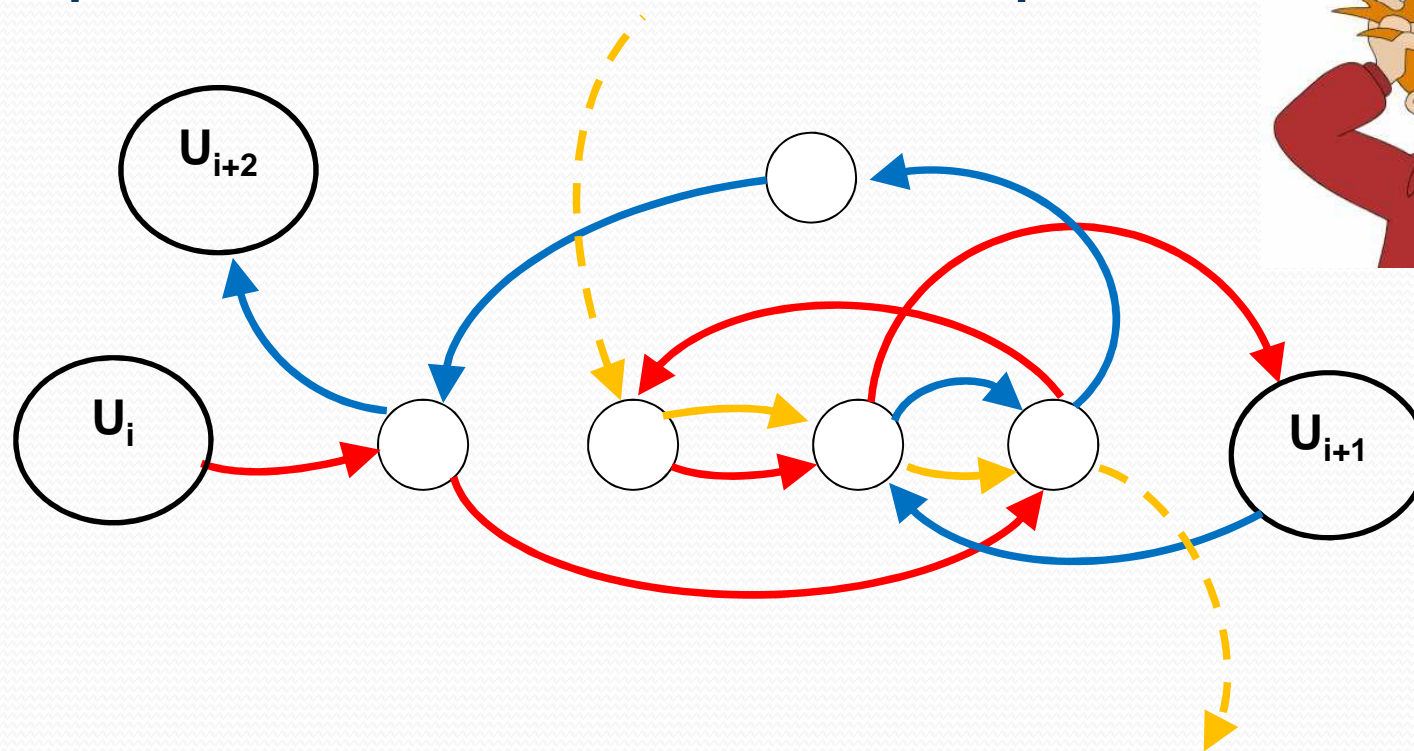
Did I say “easy”?

- **Spoke** paths of DRs have a simple tree structure
- **Rim** paths, instead, can be complex



Did I say “easy”?

- **Spoke** paths of DRs have a simple tree structure
- **Rim** paths, instead, can be complex



- When a DR has arbitrarily complex rim paths, showing an oscillation is not trivial

Some gory details

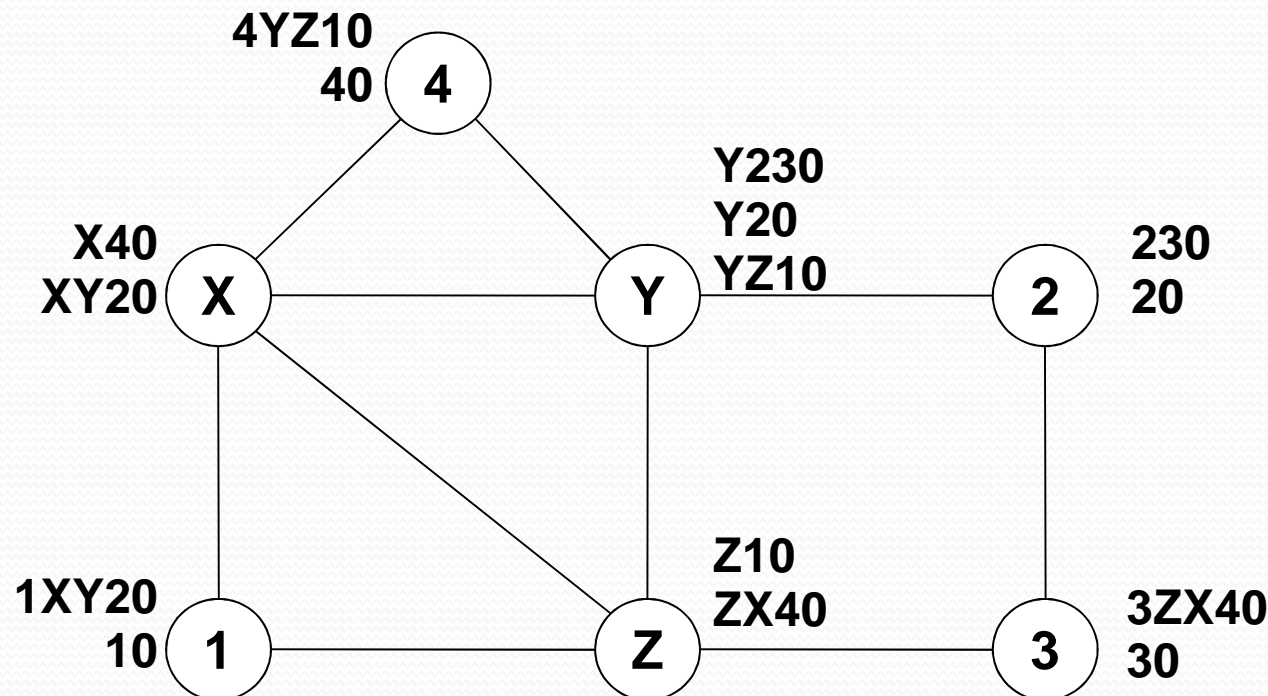
- What if a DR with 2 pivots has intersections between rim paths?
 - this prevents the synchronous oscillation!
 - Lemma (4.2): if there exists a DR with 2 pivots then there is a DR with 2 pivots and no intersections between rim paths
- What if a DR with >2 pivots is such that the announcement of a pivot cannot reach the next one (e.g., due to routing preferences)?
 - this prevents the asynchronous oscillation!
 - Lemma (4.4): in this case, we can find another DR having only 2 pivots

Multiple stable states

- Multiple stable states (e.g., BGP wedgies) imply the presence of a DR (Theorem 4.2)
 - as a consequence, the network is provably not SUF
- If we are given two stable states
 - then we can find a DW [GriffinShepherdWilfong02]
 - we show that this DW is actually a DR
 - as such, it does pose stability problems
 - that is, it is possible to pinpoint trouble points in the network (pivot nodes of the DR)
 - just by analyzing the routing tables, no need to know the routing policies
 - analysis takes linear time

Filters vs Cable Cuts

- SUF trivially implies Robustness (i.e., stability under arbitrary link faults)
- The converse does not hold
 - this network ("Filthy-Gadget") is robust but is not SUF
 - misconfigured filters are worse than cable cuts!



Conclusions

- Step forward in BGP formal analysis
 - Characterization for SUF
 - SUF bound to a structure of routing policies
 - such a property does not depend on network dynamics
 - Robustness does not imply SUF
 - NOCs might be more dangerous than scissors
- Practical implication
 - Easy to find trouble points given multiple stable states
 - without knowledge of the policies
- Future Work
 - How hard is finding a DR in a BGP configuration?
 - Implications on iBGP?