The *Stable Paths Problem* As A Model Of BGP Routing

Timothy G. Griffin
AT&T Labs
griffin@research.att.com

Joint work with

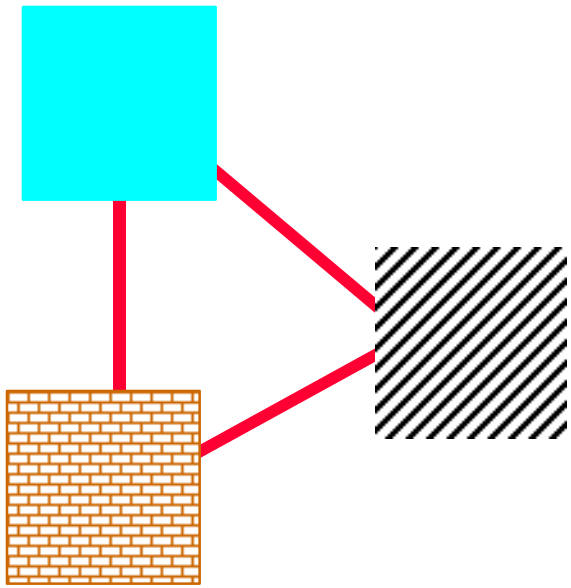
Bruce Shepherd
Bell Labs
bshep@research.bell-labs.com

Gordon Wilfong
Bell Labs
gtw@research.bell-labs.com

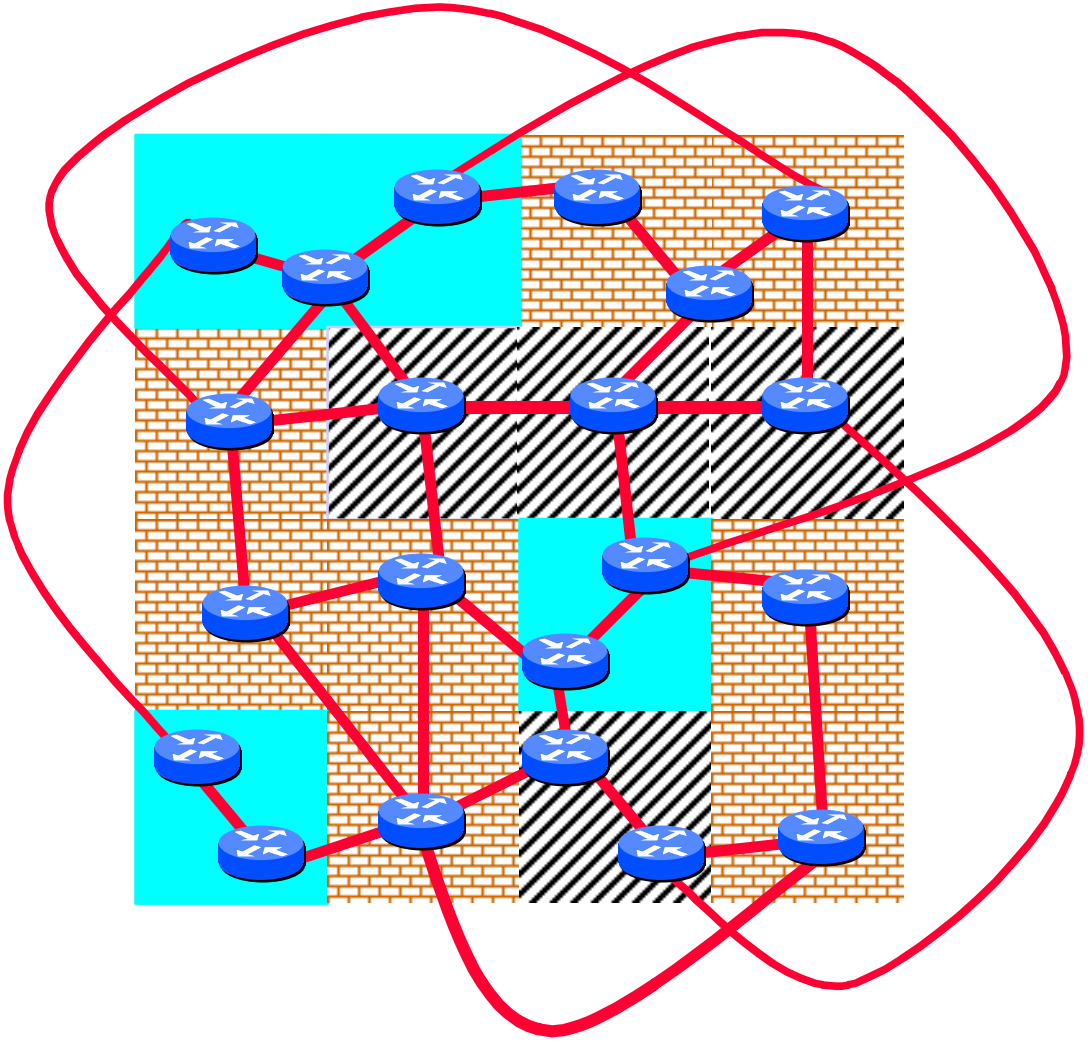
Leiden Global Internet Workshop

September, 2000

AS graphs obscure topology!



The AS graph may look like this.



Reality may be closer to this…

Interior vs. Exterior Routing Protocols

Interior Gateway Protocols (IGP) : inside autonomous systems

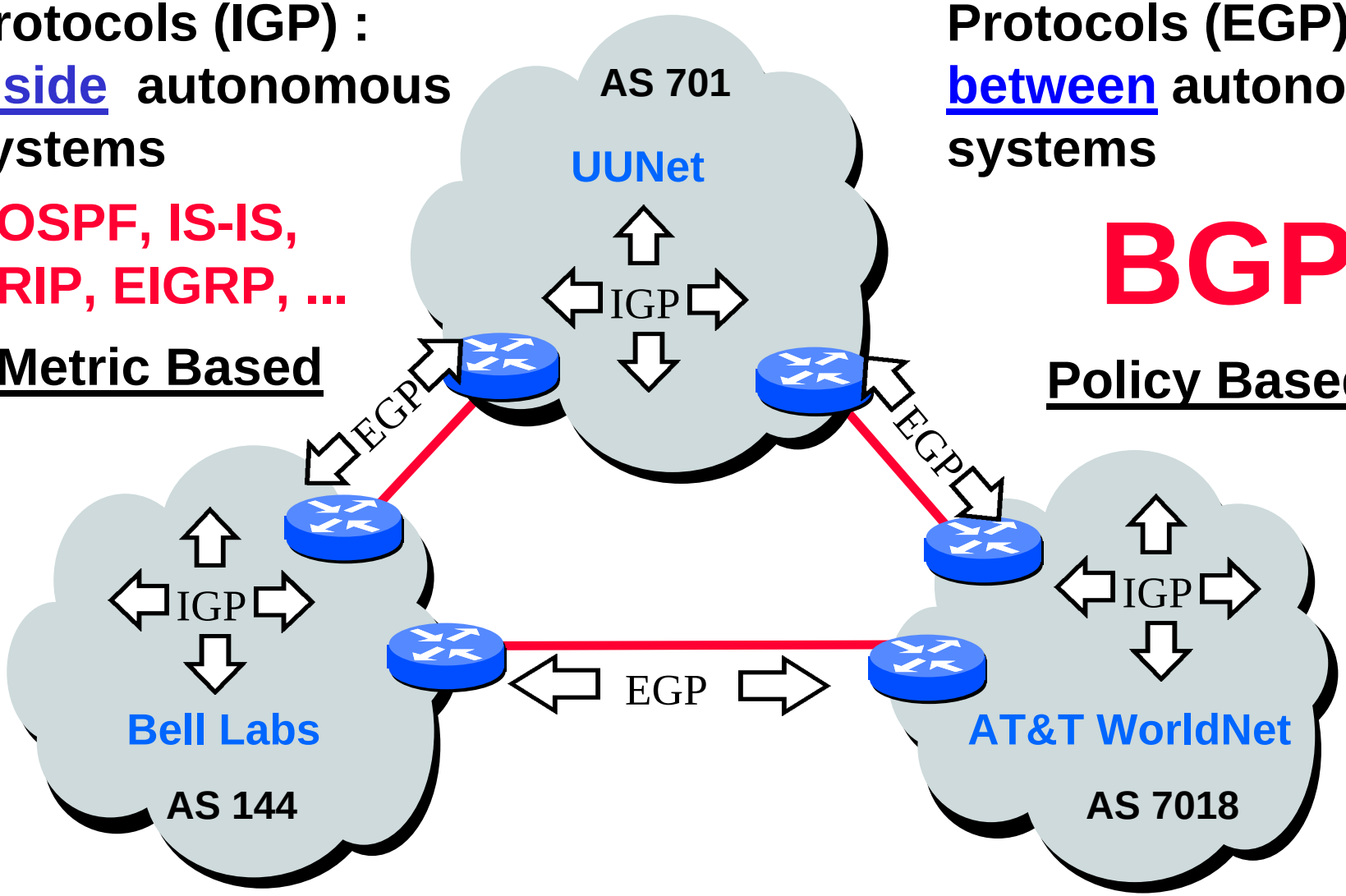
OSPF, IS-IS, RIP, EIGRP, ...

Metric Based

Exterior Gateway Protocols (EGP) : between autonomous systems

BGP

Policy Based



Starting point : BGP can diverge

- **BGP is not guaranteed to converge to a stable routing. Policy inconsistencies can lead to "livelock" protocol oscillations.**
- **See "Persistent Route Oscillations in Inter-domain Routing" by K. Varadhan, R. Govindan, and D. Estrin. ISI report, 1996**

Goal : A simple, tractable, and complete model of BGP routing

One application : sufficient conditions that guarantee no divergence.

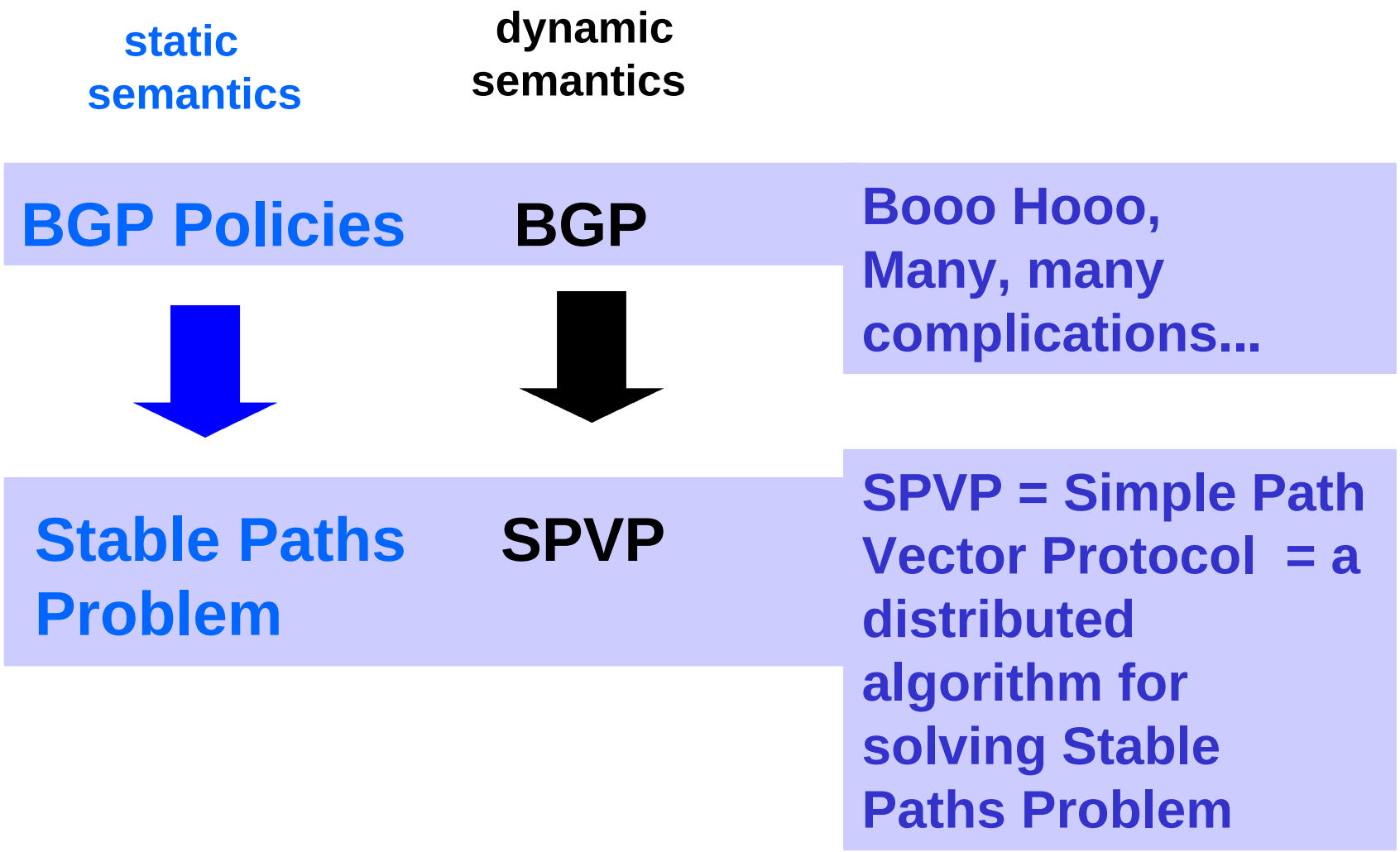
What Problem is BGP solving?

Underlying problem	Distributed means of computing a solution.
Shortest Paths	RIP, OSPF, IS-IS
X?	BGP

Having an X can

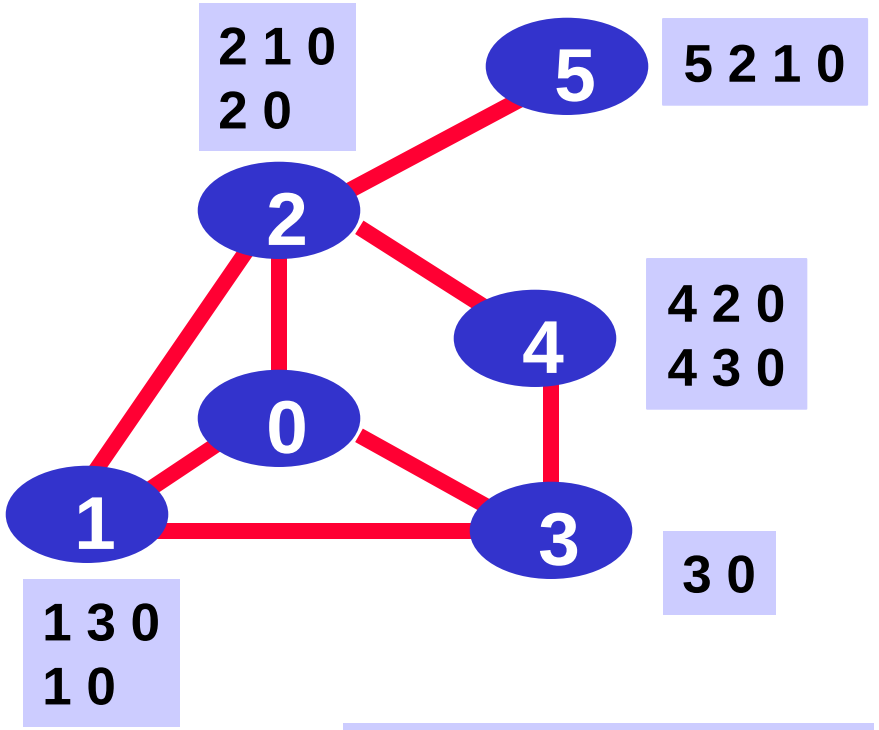
- aid in the design of policy analysis algorithms and heuristics,
- aid in the analysis and design of BGP and extensions,
- help explain some BGP routing anomalies,
- provide a fun way of thinking about the protocol

Separate dynamic and static semantics



An instance of the *Stable Paths Problem* (SPP)

- A graph of nodes and edges,
- Node 0, called *the origin*,
- For each non-zero node, a set or permitted paths to the origin. This set always contains the "null path".
- A ranking of permitted paths at each node. Null path is always least preferred. (Not shown in diagram)



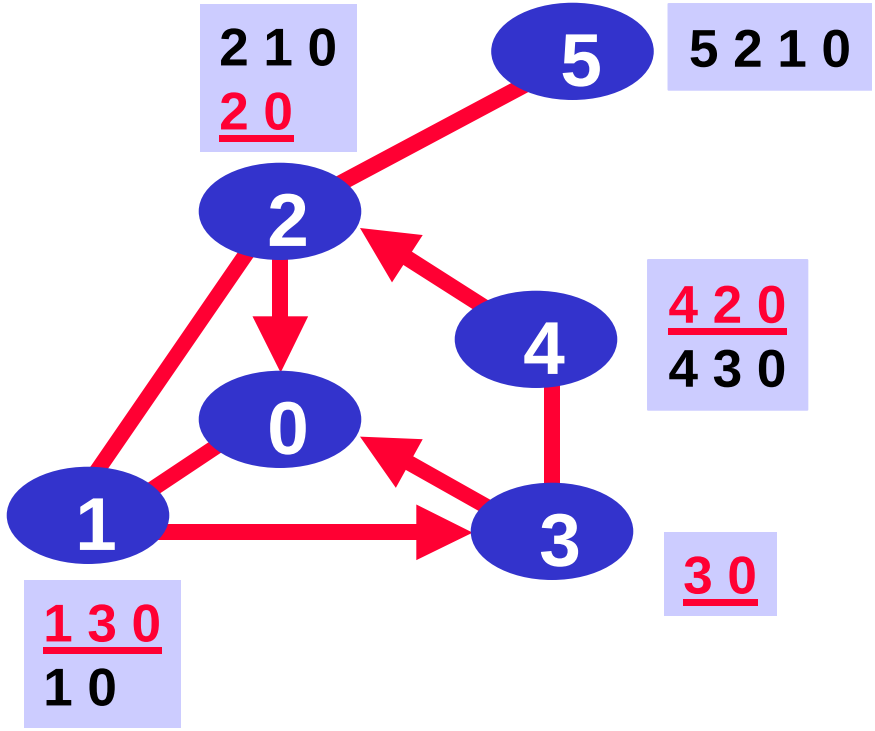
When modeling BGP : nodes represent BGP speaking routers, and 0 represents a node originating some address block

Yes, the translation gets messy!

A Solution to a Stable Paths Problem

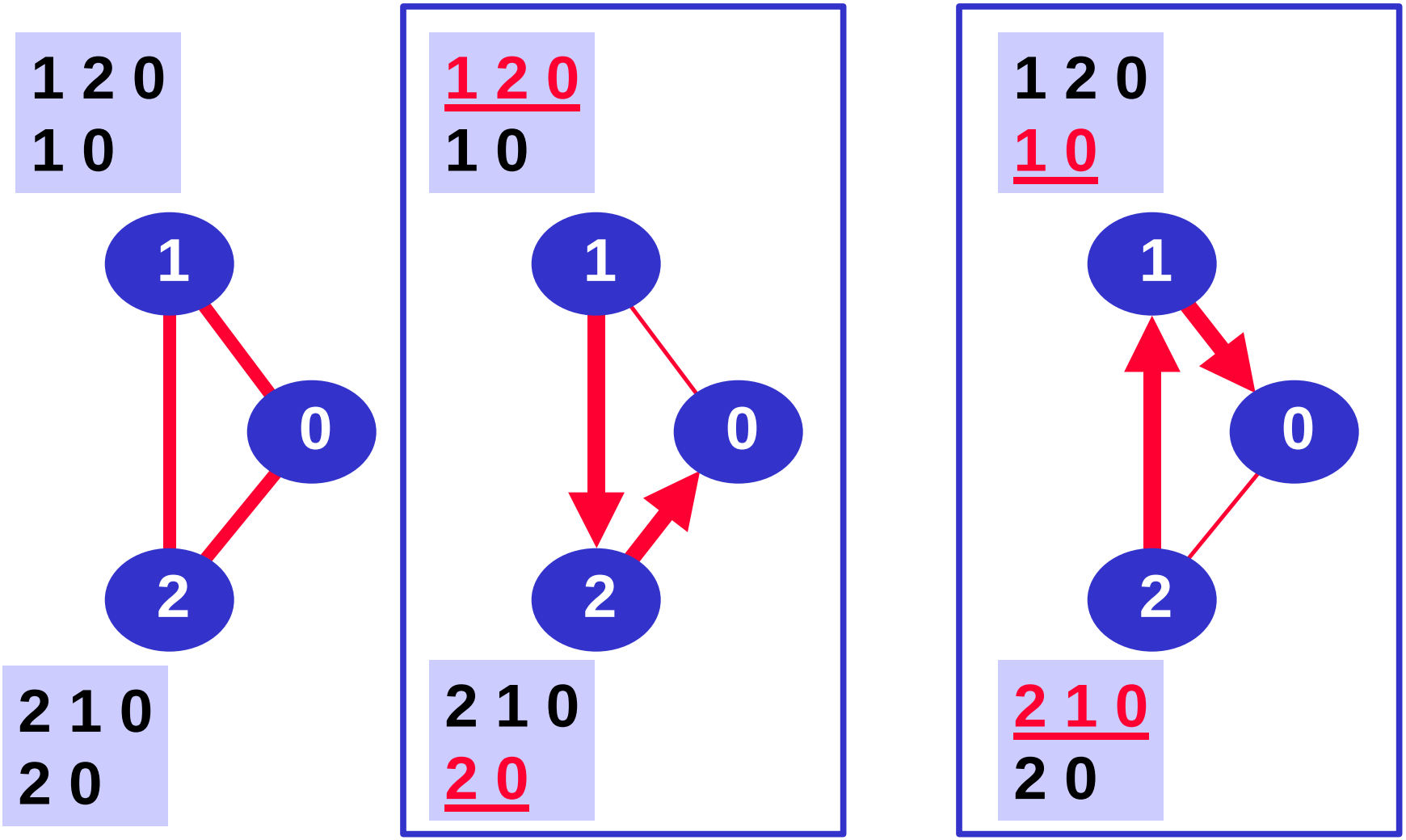
A *solution* is an assignment of permitted paths to each node such that

- **node u's assigned path is either the null path or is a path uwP, where wP is assigned to node w and {u,w} is an edge in the graph,**
- **each node is assigned the highest ranked path among those consistent with the paths assigned to its neighbors.**



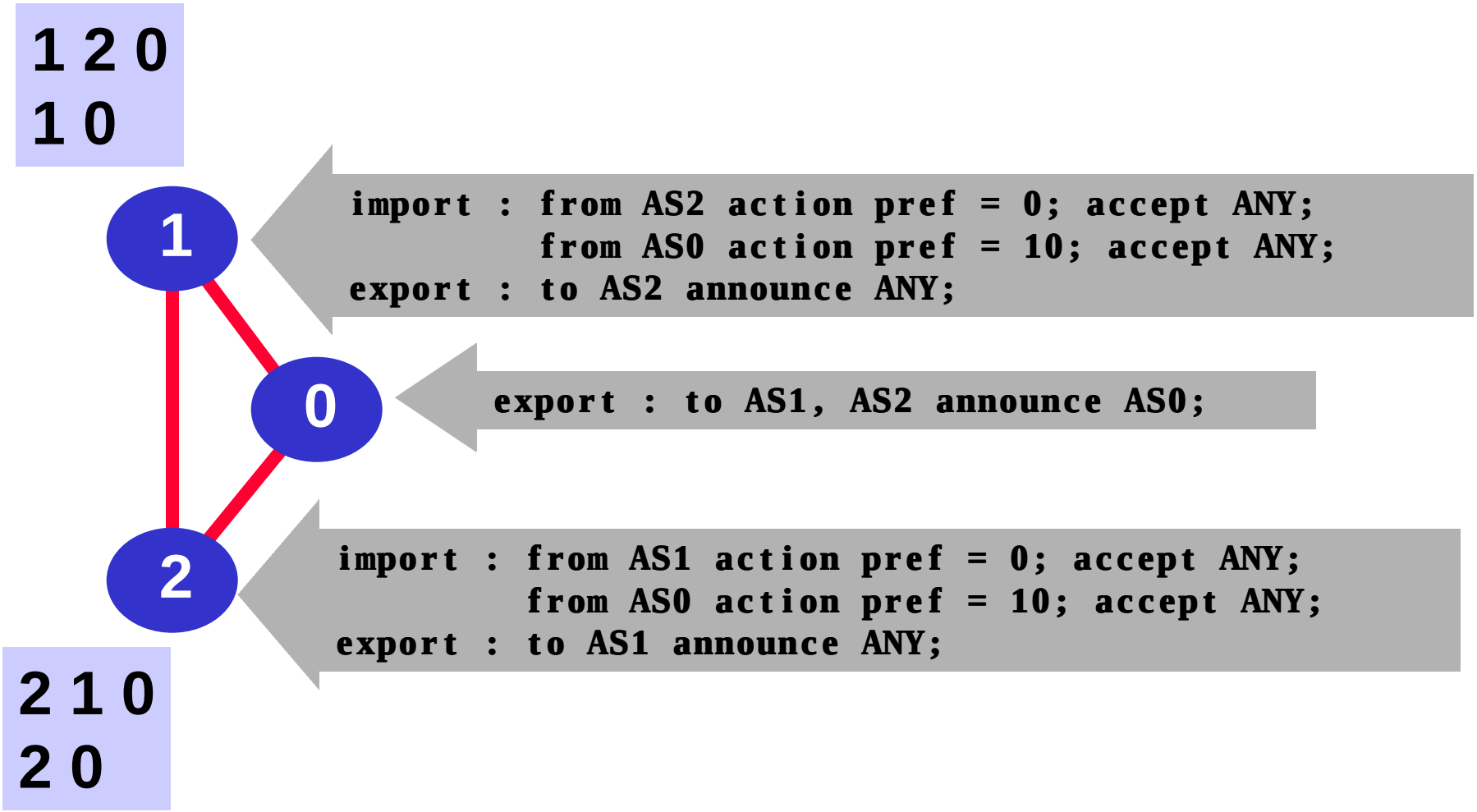
A Solution need not represent a shortest path tree, or a spanning tree.

An SPP may have multiple solutions

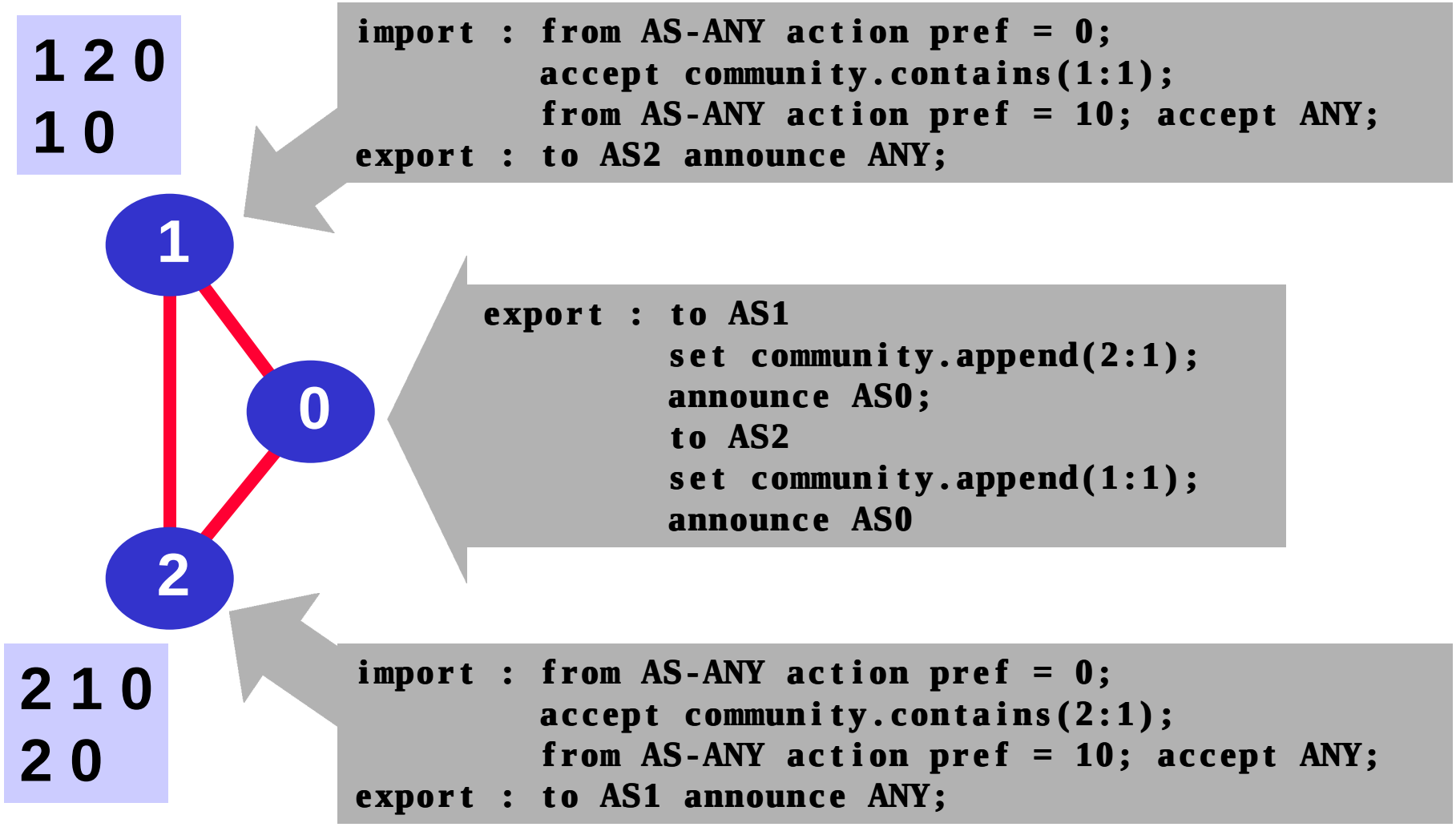


Mapping BGP routing policies down to a Stable Paths Problem :

DISAGREE in RPSL (Version I)



DISAGREE in RPSL (Version II)

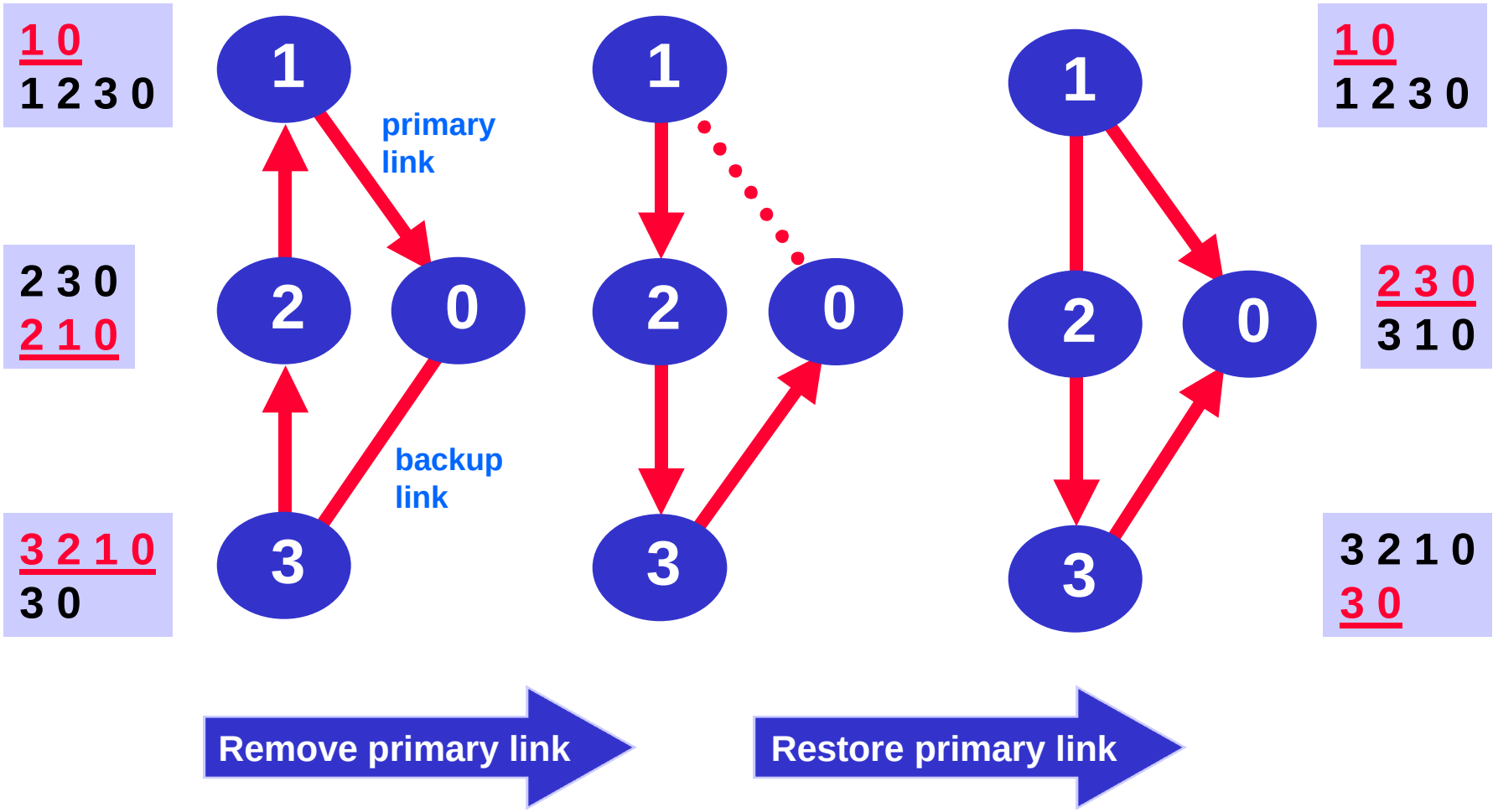


Assume AS1 and AS2 use "`neighbor send-community`" command ....

DISAGREE in RPSL (Version III)

The interaction of all BGP policies is directly represented in SPP

Multiple solutions can result in "Route Triggering"



Ranking paths

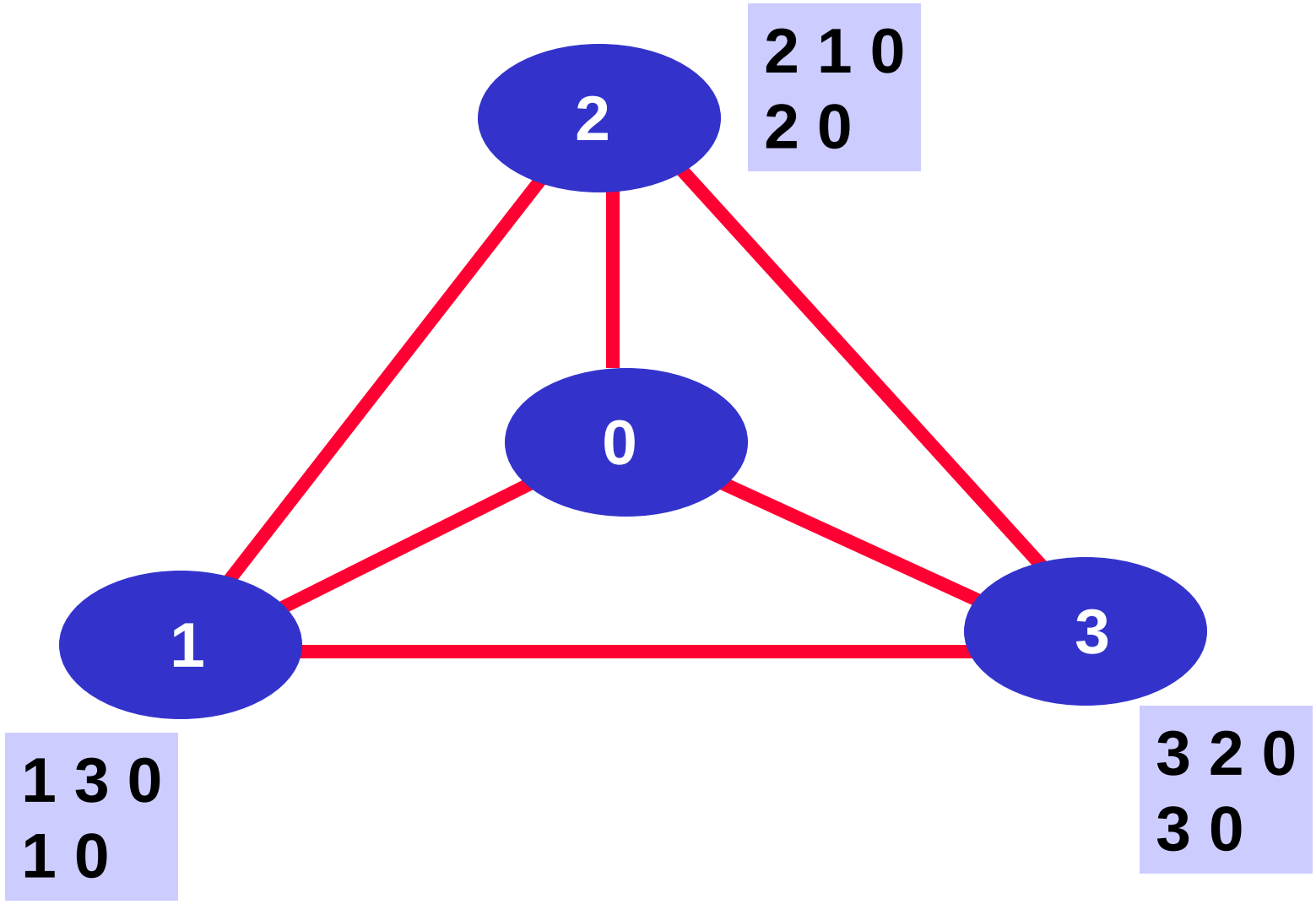
(lp, 1/(pl+1), origin, 1/(med+1), type, 1/(igp+1), nh)

Rank r

Lexicographic order

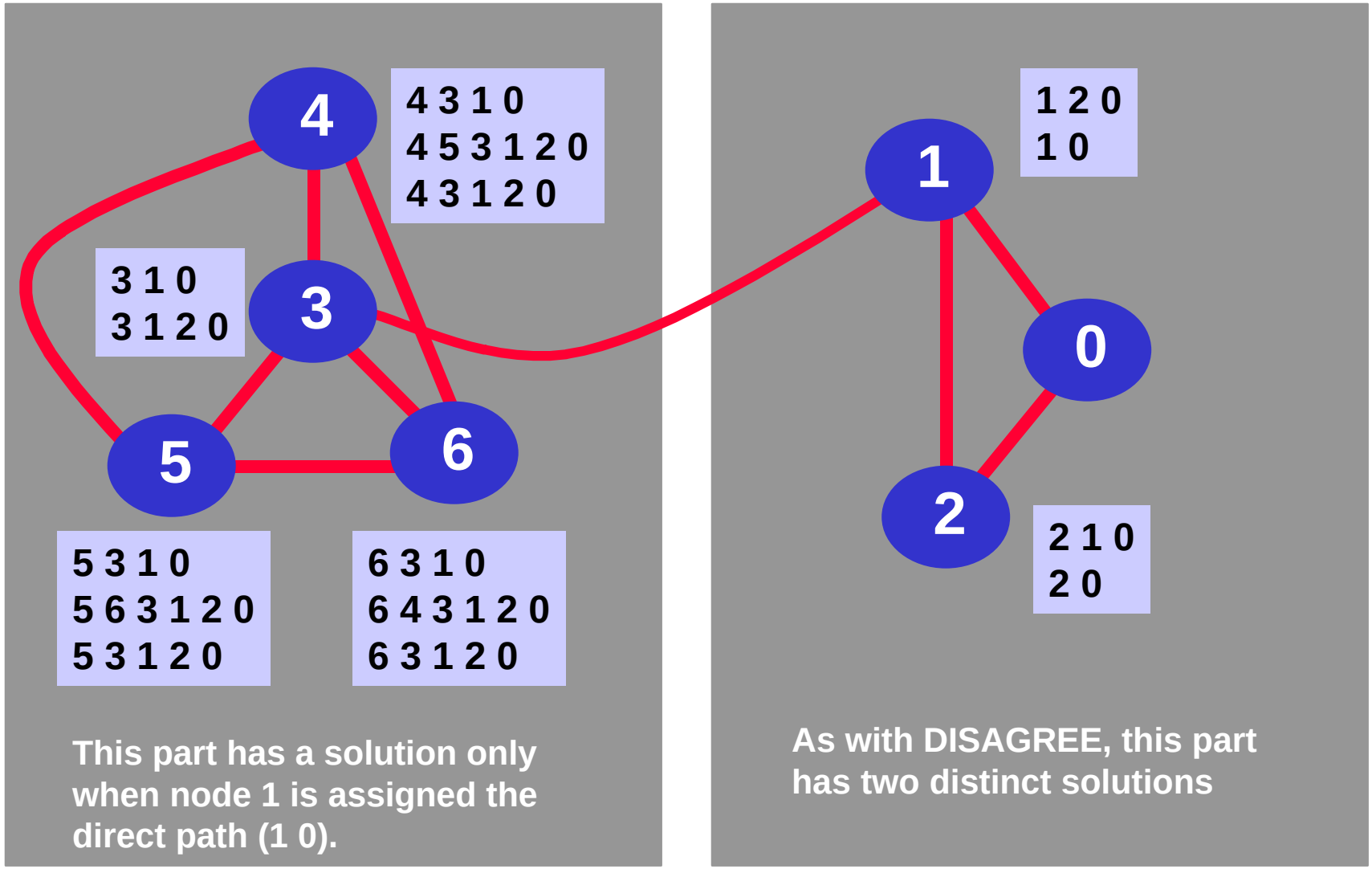
lp = local preference
pl = as-path length
origin in {IGP, EGP, INCOMPLETE}, IGP < EGP < INCOMPLETE
type in {IBGP, EBGP}, IBGP < EBGP
igp = interior routing metric
nh = next hop ip address

BAD GADGET : No Solution



PRECARIOUS

Has a solution, but can get "trapped"



Solving an SPP

Just enumerate all path assignments And check stability of each….

!! Exponential complexity !!

But, in worst case you (probably) can’t do any better…

Remember 3-SAT?

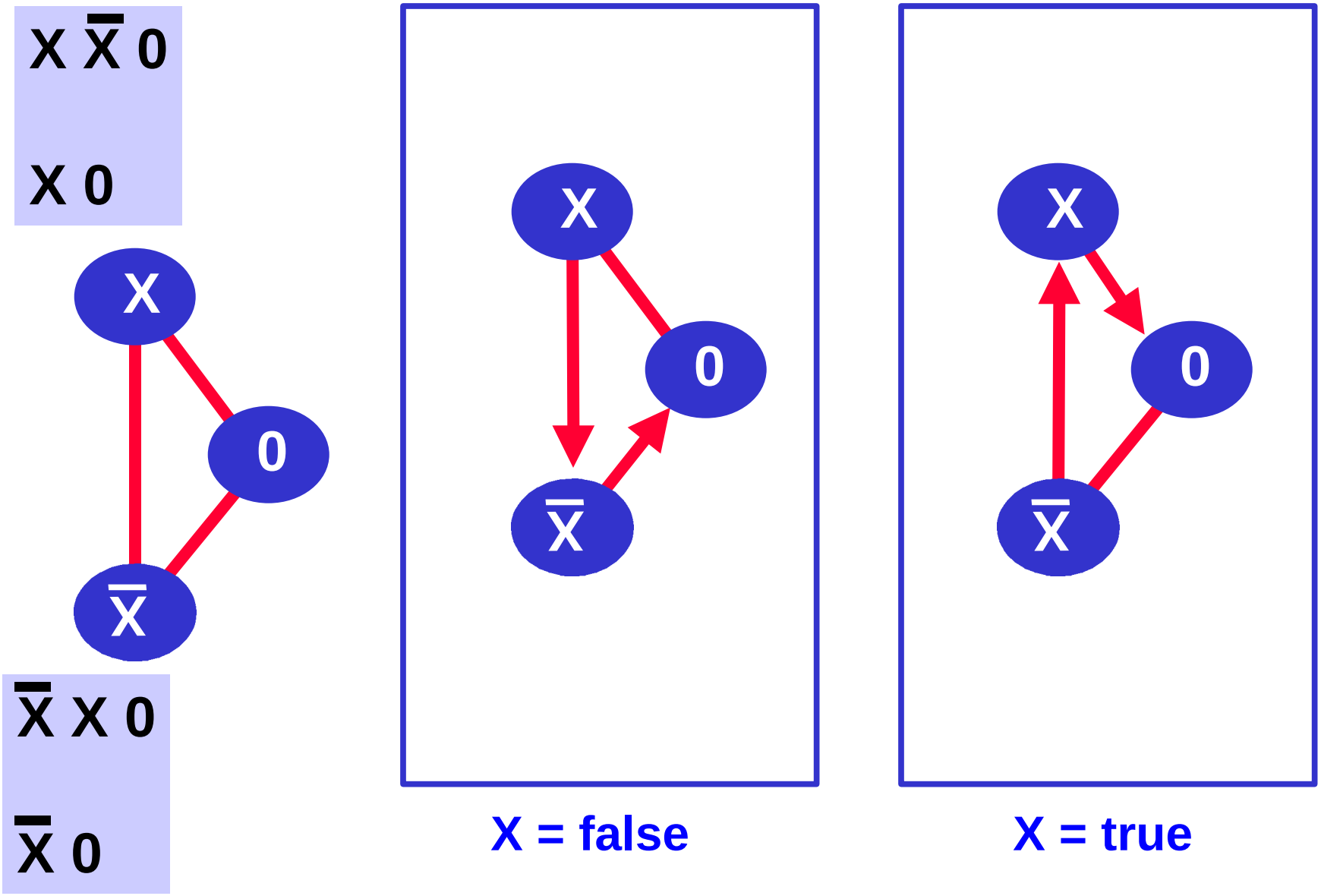
Variables V = {X1, X2, …, Xn}

Clauses C1 = X17 or ~X23 or ~X3,
C2 = ~X2 or X3 or ~X12
….
Cm = X6 or ~X7 or X18

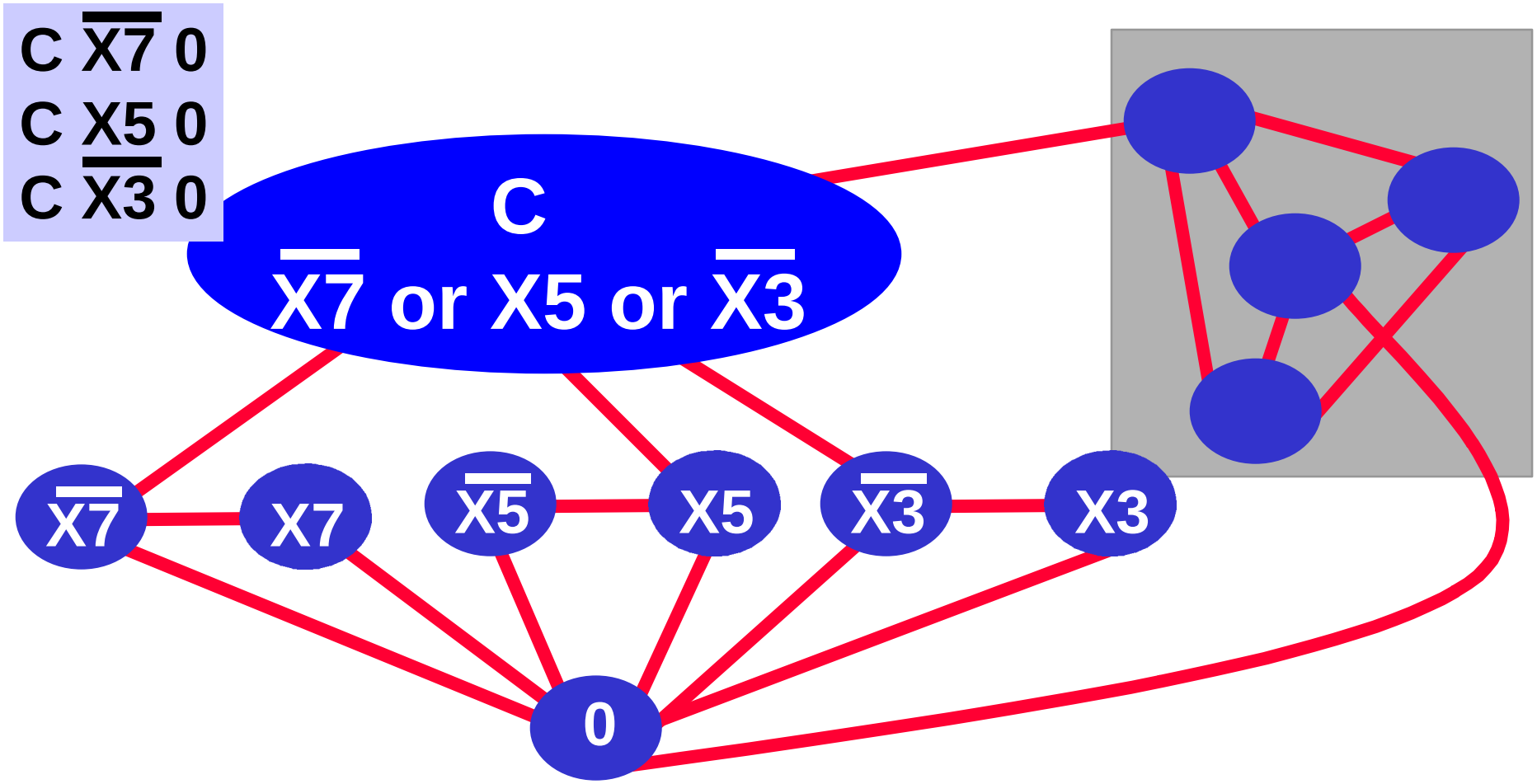
Question Is there an variable assignment
A : V → {true, false} such that
each clause C1, … ,Cm is true?

3-SAT is NP-complete

Modeling assignment to variable X



Solvability is NP-complete



Distributed algorithms to solve SPP?

- **OSPF-like :**
  - **Distribute topology, path ranks**
  - **Solve SPP locally**
  - **Exponential worst case**
  - **How can loops be avoided when multiple solutions exist?**
- **RIP-like:**
  - **Pick the best path from the set of your neighbor's paths, tell your neighbors when you change your mind**
  - **Can diverge**
  - **Not guaranteed to find a solution, even when one exists**
  - **No bound on convergence time**

SPVP protocol

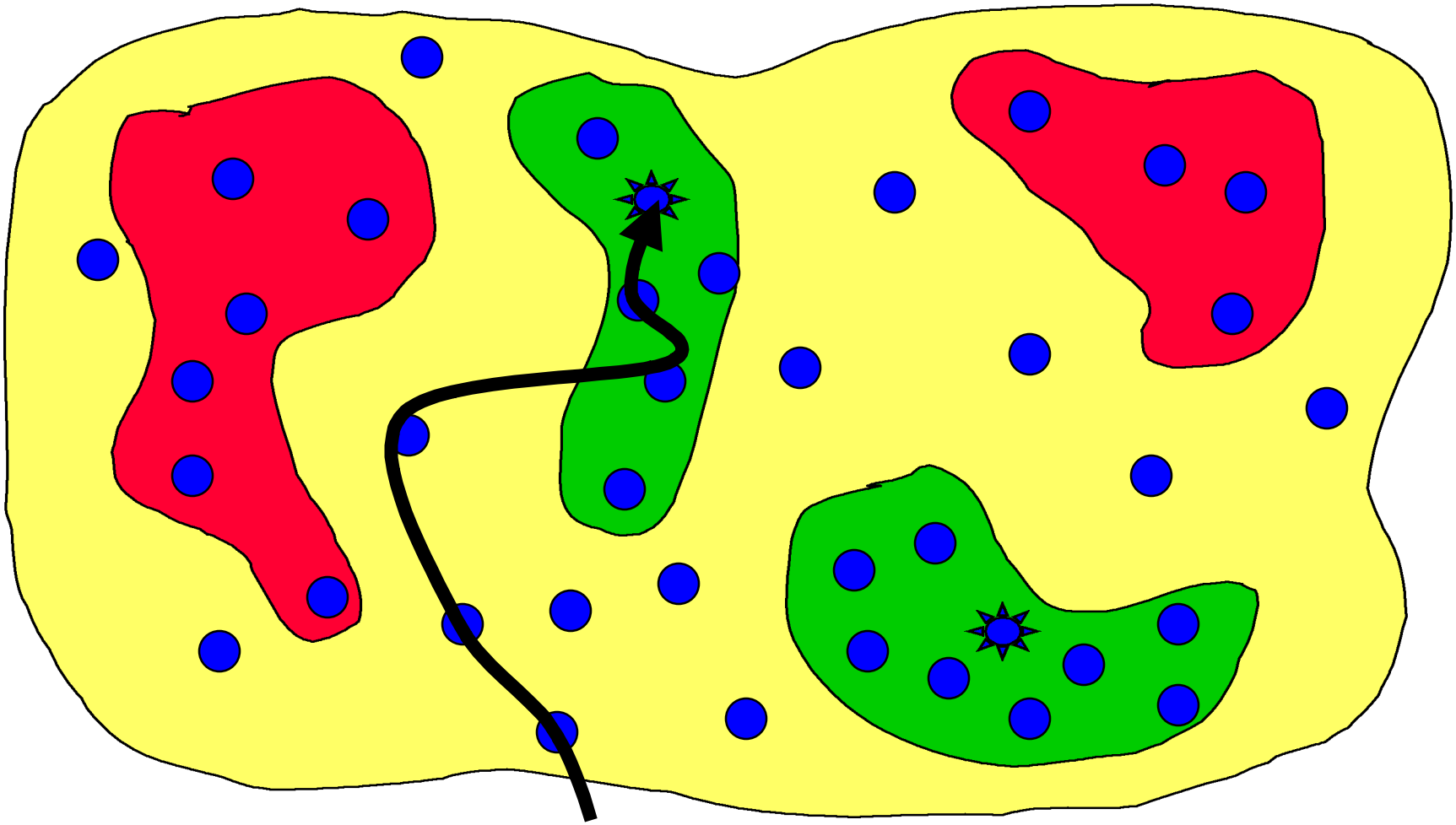
Pick the best path available at any given time…

```
process spvp[u]
{
   receive P from w →
   {  rib-in(u←w) := u P
      if rib(u) != best(u) {
         rib(u) := best(u)
         foreach v in peers(u) {
            send rib(u) to v
         }
      }
   }
}
```

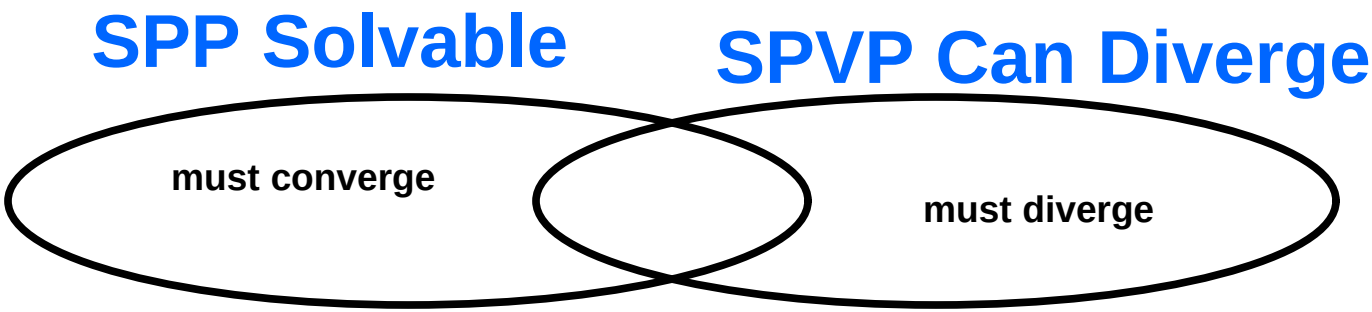
SPVP wanders around assignment space

= assignment

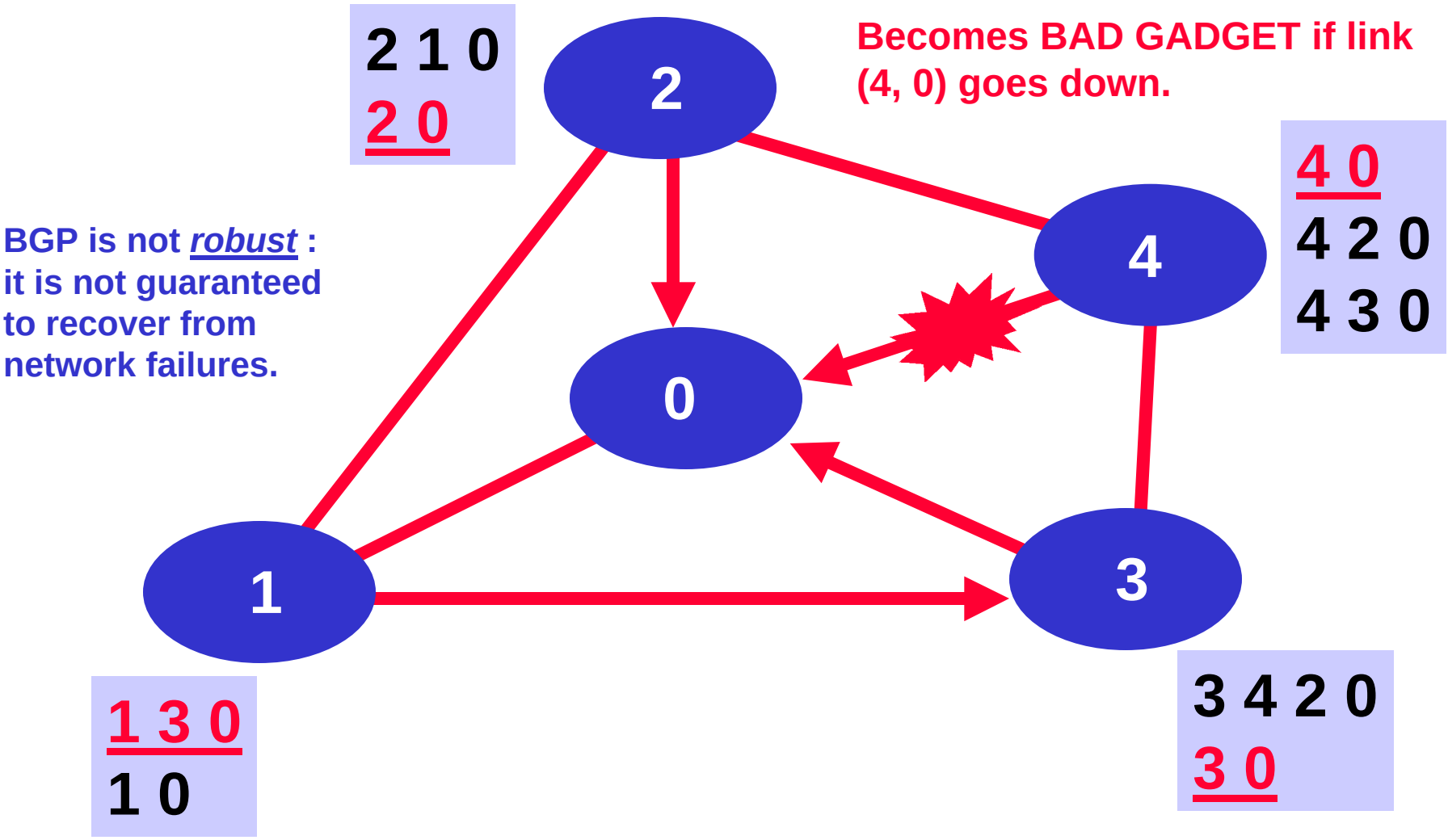
= solution



Another Picture



SURPRISE : Beware of Backup Policies



SPP simplifies analysis of proposed extensions

Proposal : Allow AS-path length to include member ASes within a confederation.

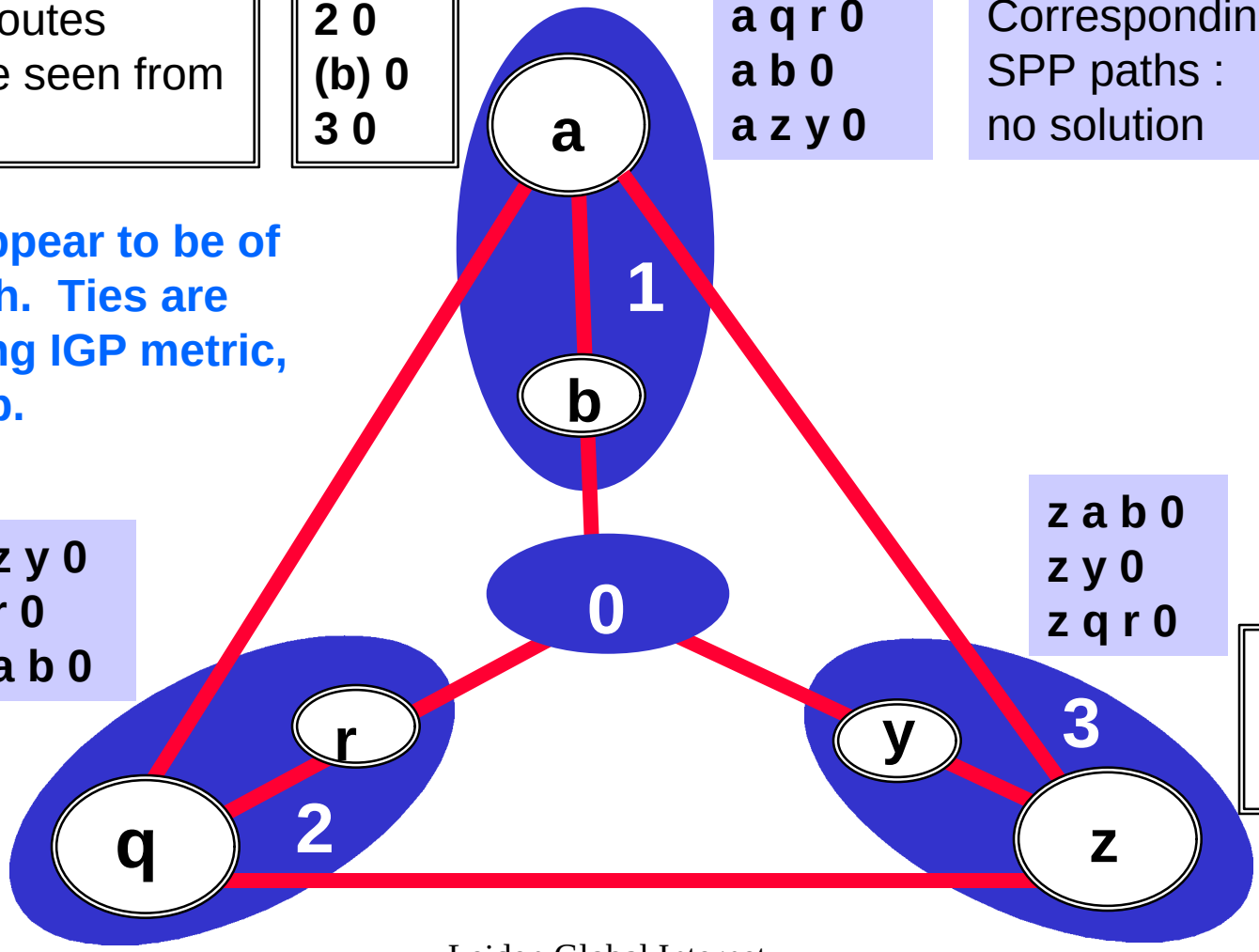
AS path of routes in BGP table seen from member-AS

2 0
(b) 0
3 0

a q r 0
a b 0
a z y 0

Corresponding SPP paths : no solution

All paths appear to be of equal length. Ties are broken using IGP metric, or Next-Hop.



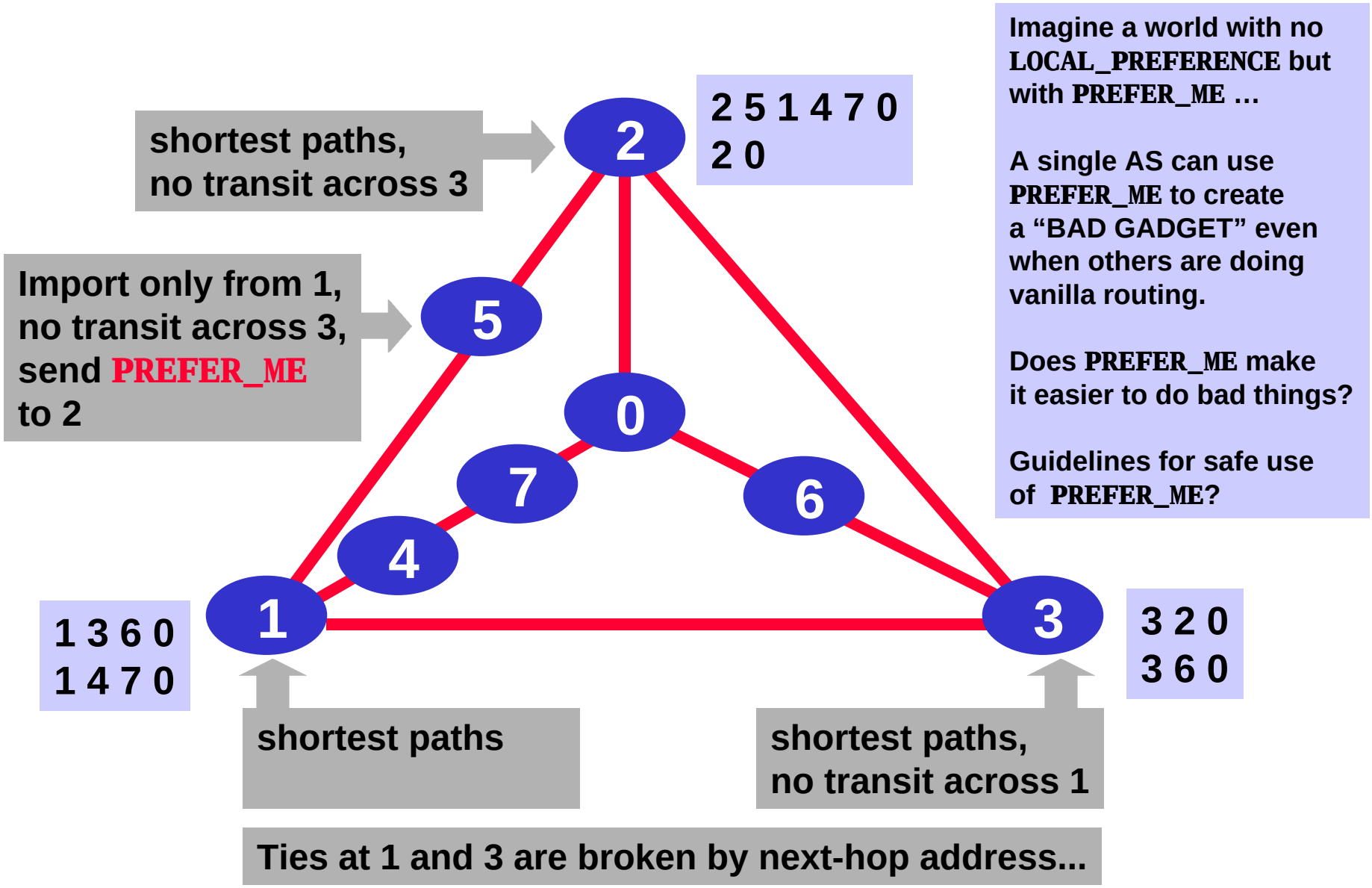
q z y 0
q r 0
q a b 0

z a b 0
z y 0
z q r 0

3 0
(r) 0
1 0

1 0
(y) 0
2 0

PREFER_ME Global Community?



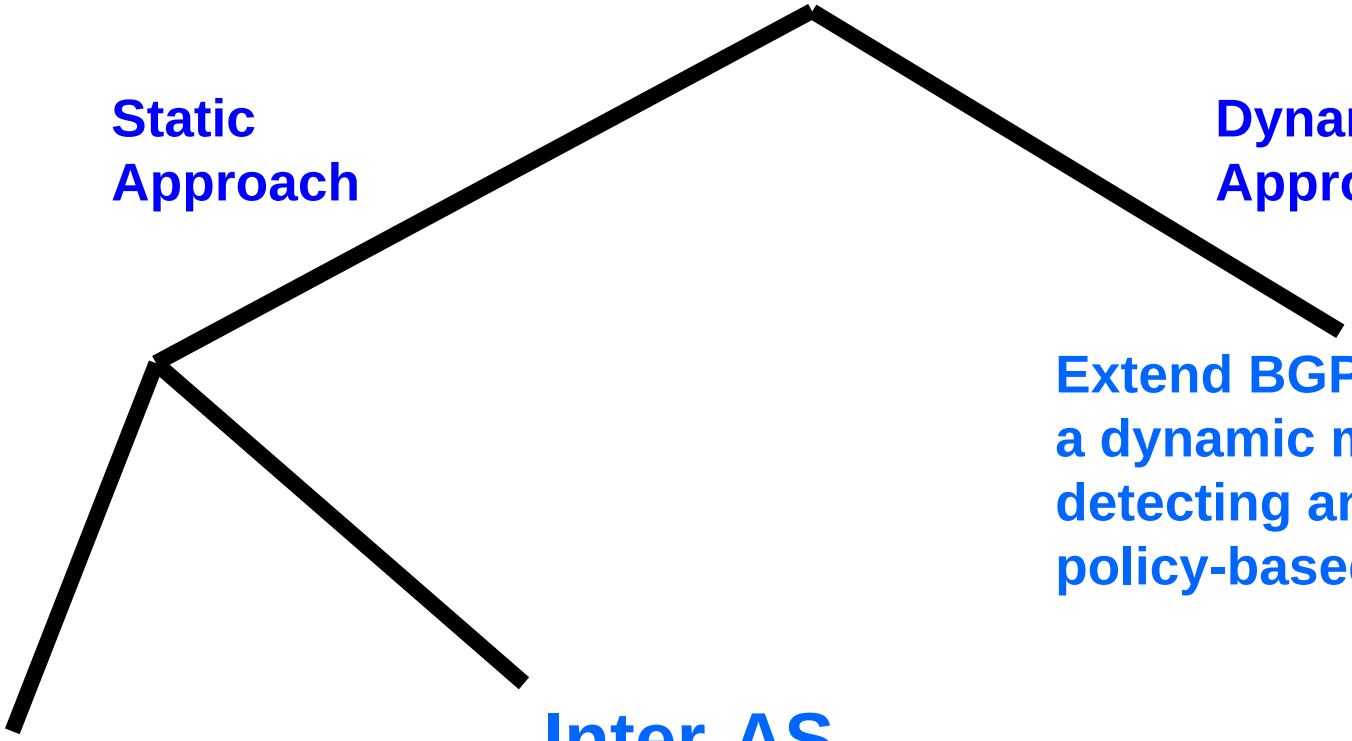
Imagine a world with no LOCAL_PREFERENCE but with PREFER_ME …

A single AS can use PREFER_ME to create a "BAD GADGET" even when others are doing vanilla routing.

Does PREFER_ME make it easier to do bad things?

Guidelines for safe use of PREFER_ME?

What is to be done?



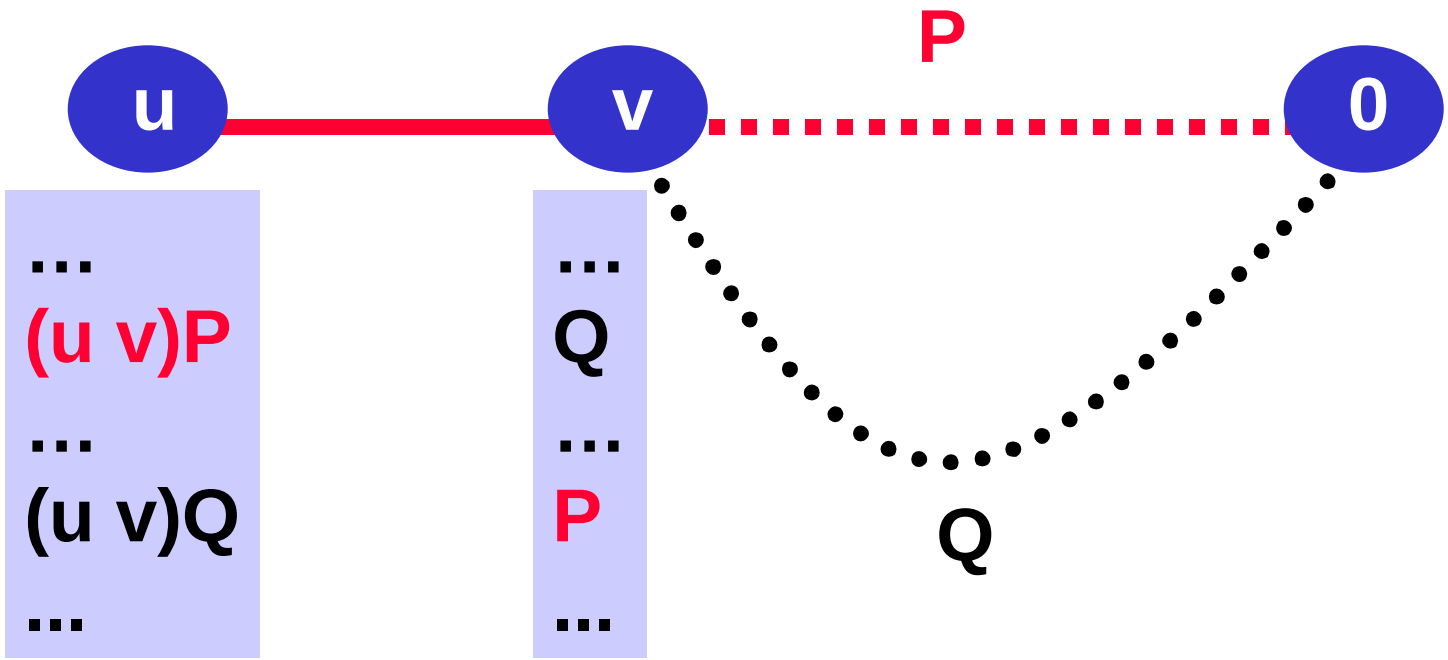
These approaches are <u>complementary</u>

A sufficient condition for sanity

If an instance of SPP has an acyclic dispute digraph, then

Static (SPP)	Dynamic (SPVP)
solvable	safe (can't diverge)
unique solution	predictable restoration
all sub-problems uniquely solvable	robust with respect to link/node failures

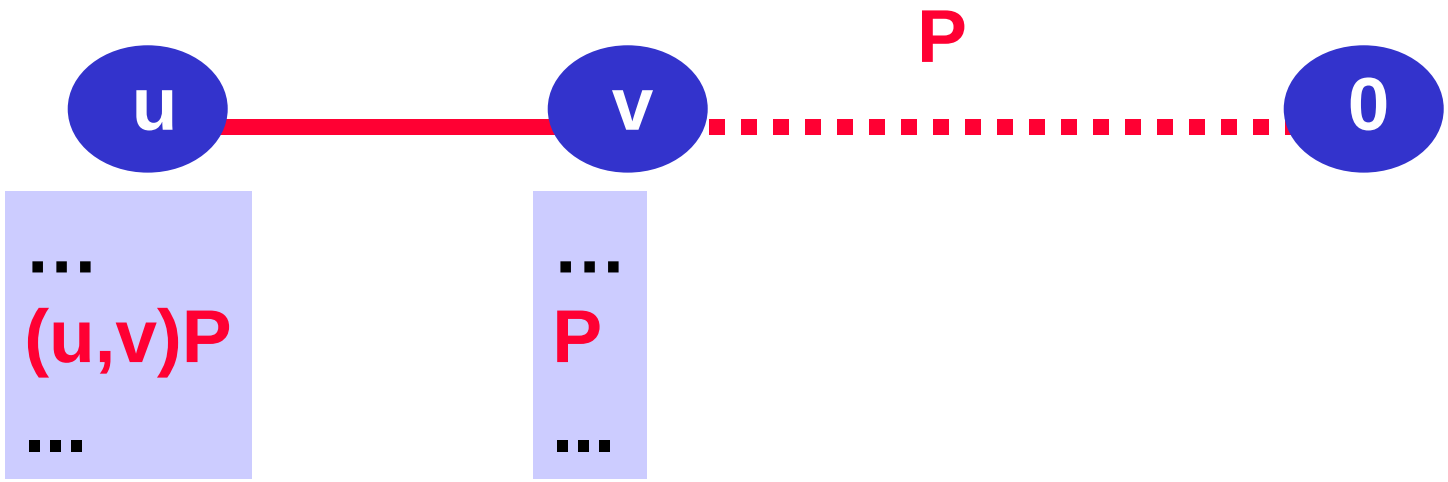
Dispute Digraph



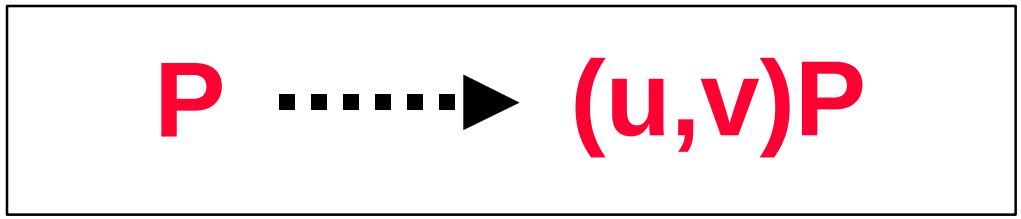
Gives the *dispute arc*

$$Q \longrightarrow (u\ v)P$$

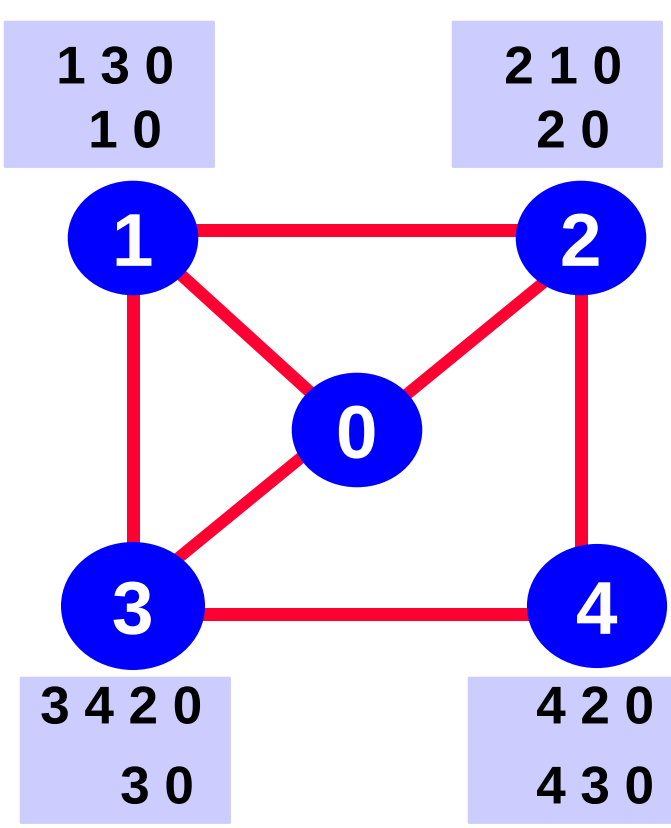
Dispute Digraph (cont.)

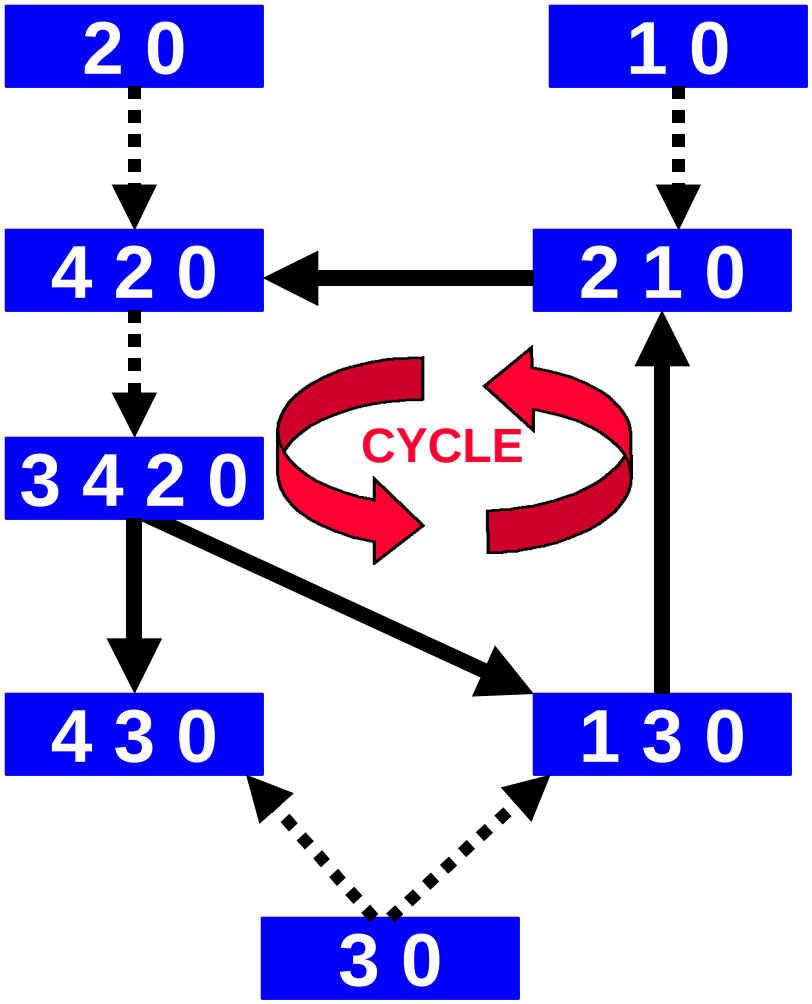


Gives the *transmission arc*



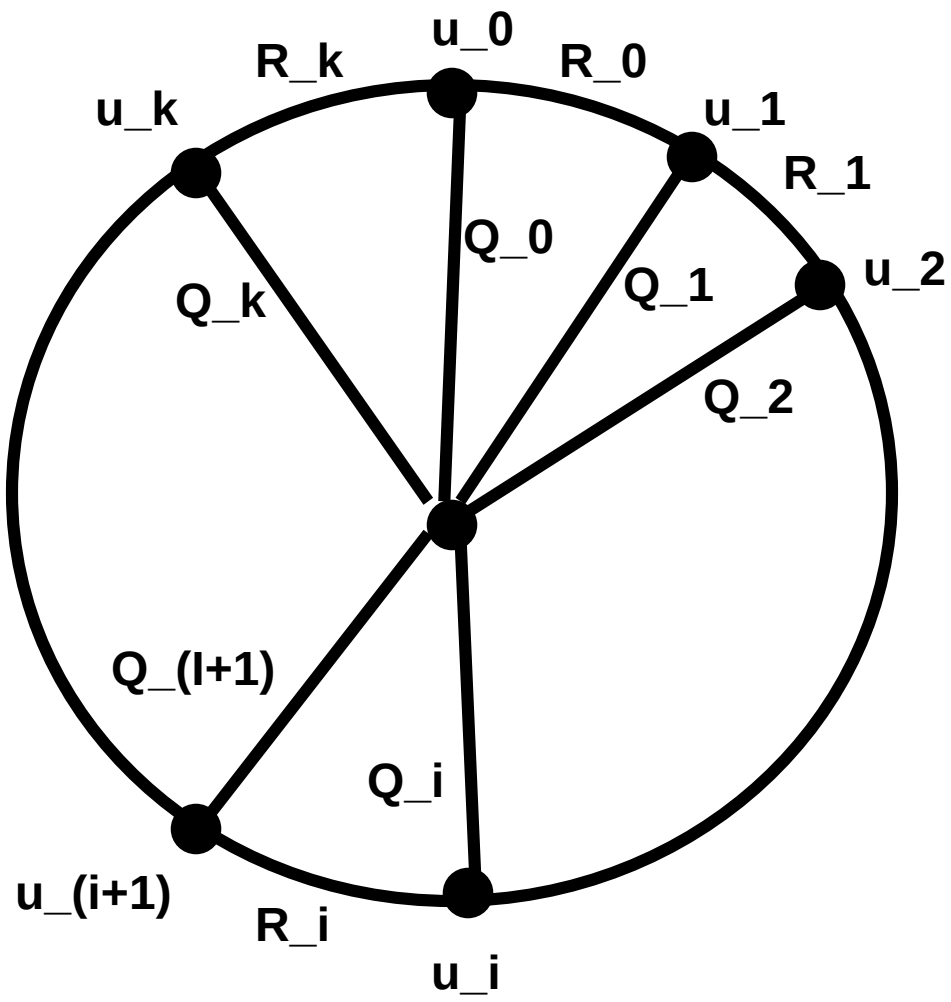
Dispute Digraph Example





BAD GADGET II

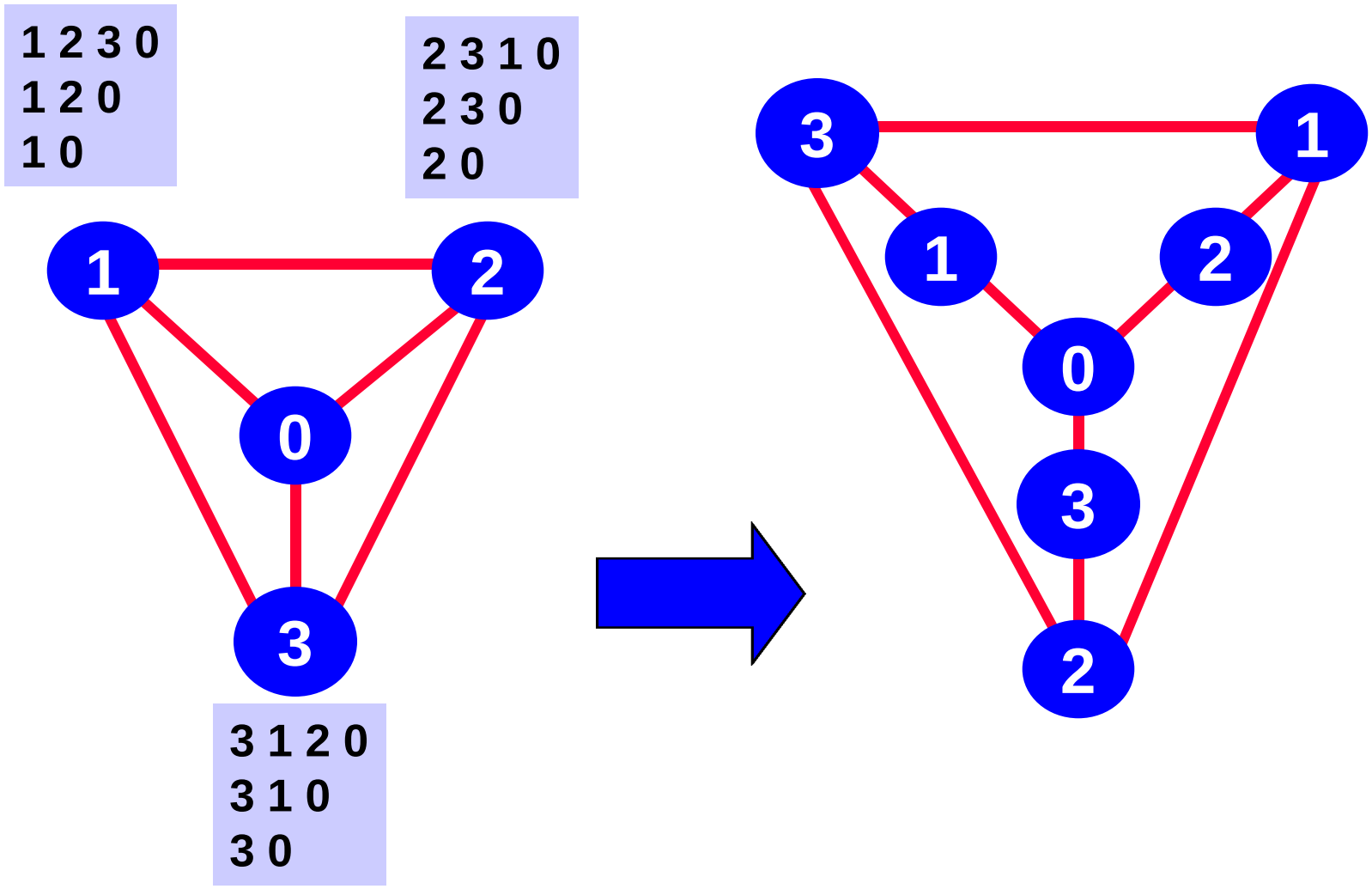
Dispute Wheels



At u_i, rank of Q_i is less than or equal to rank of R_iQ_(i+1)

There exists a dispute wheel iff there exists cycle in the dispute digraph

Dispute Wheel Example

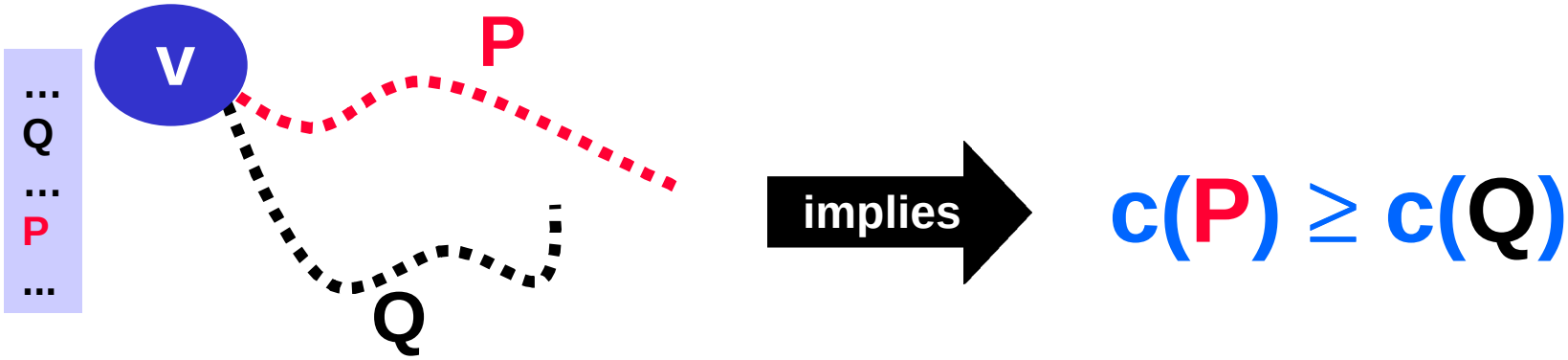


An Application

c is a cost function on edges.

c is *coherent* if all cycles have positive cost.

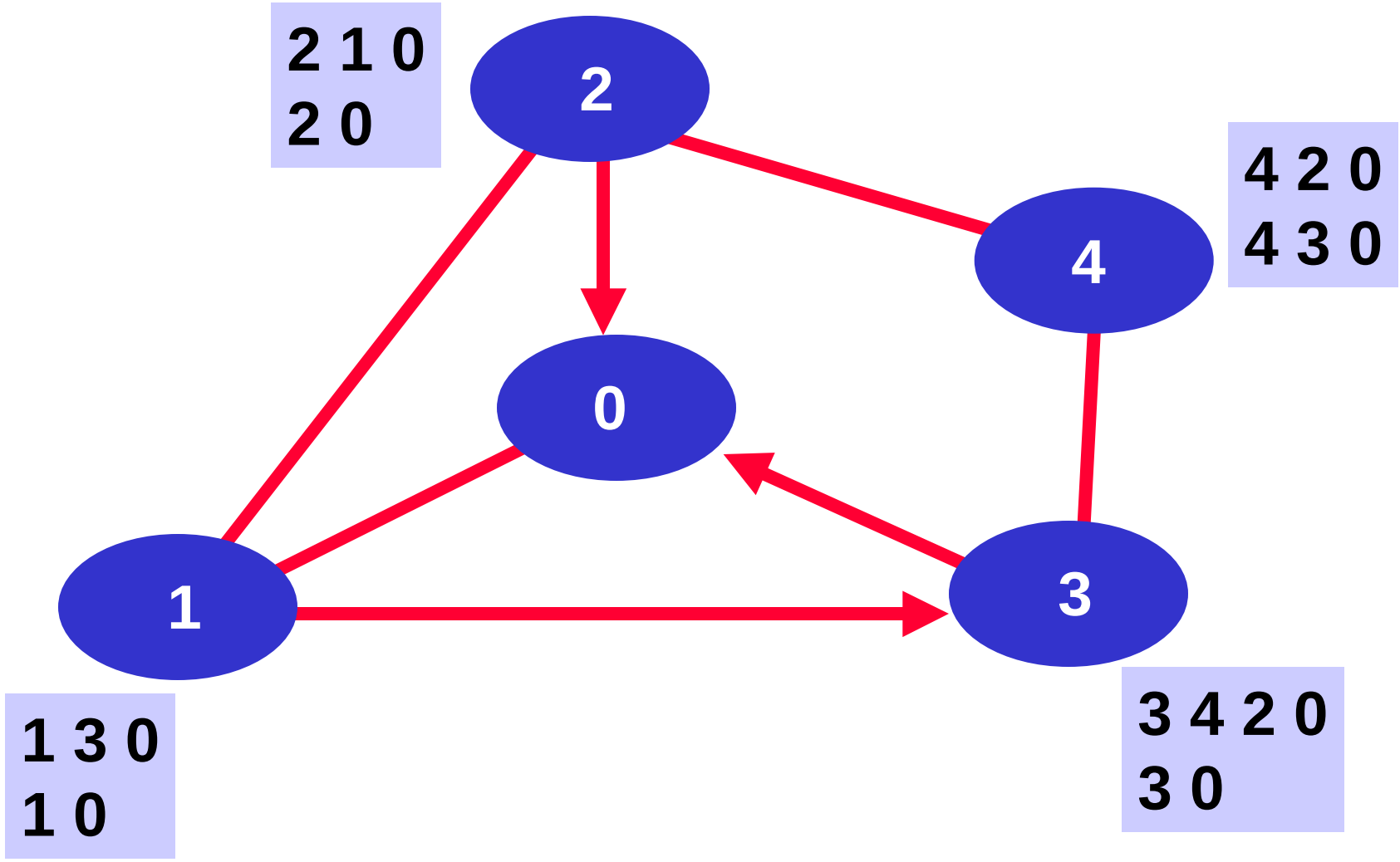
An SPVP specification is *consistent with c* if



A Dynamic Solution

- **Extend SPVP with a history attribute,**
- **A route's history contains a path in the dispute digraph that "explains" how the route was obtained,**
- **A route history will contain a dispute cycle if and only if a policy dispute is dynamically realized.**
- **If a route's history contains a cycle, then suppress it ….**

BAD GADGET II



A History Trace for Bad Gadget II

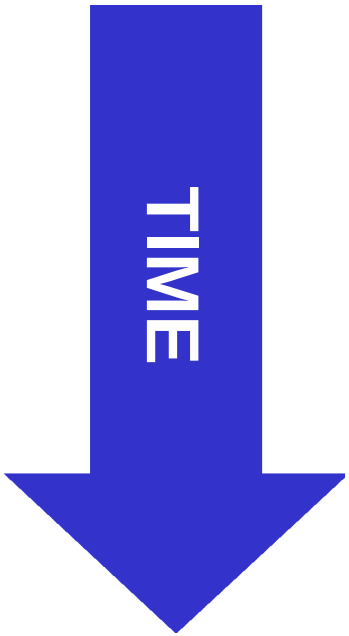
path	event history for path
1 : (1 0)	e
2 : (2 0)	e
3 : (3 4 2 0)	e
4 : (4 2 0)	e

▪ ▪ ▪ ▪ ▪ ▪ ▪ ▪ ▪ ▪ ▪ ▪

1 : (1 0)	e
2 : (2 1 0)	(+ 2 1 0)
3 : (3 0)	(- 3 4 2 0) (- 4 2 0) (+ 2 1 0)
4 : ( )	(- 4 2 0) (+ 2 1 0)

1 : (1 3 0)	(+ 1 3 0) (- 3 4 2 0) (- 4 2 0) (+ 2 1 0)
2 : (2 1 0)	(+ 2 1 0)
3 : (3 0)	(- 3 4 2 0) (- 4 2 0) (+ 2 1 0)
4 : ( )	(- 4 2 0) (+ 2 1 0)

1 : (1 3 0)	(+ 1 3 0) (- 3 4 2 0) (- 4 2 0) (+ 2 1 0)
2 : (2 0)	(- 2 1 0) (+ 1 3 0) (- 3 4 2 0) (- 4 2 0) (+ 2 1 0)
3 : (3 0)	(- 3 4 2 0) (- 4 2 0) (+ 2 1 0)
4 : ( )	(- 4 2 0) (+ 2 1 0)



A CYCLE!

▪ ▪ ▪ ▪ ▪ ▪

What's going on ?

Dynamic cycles of event history correspond exactly to static cycles in the dispute digraph.

To Do

- **Implementation of Histories (in SSFNET?)**
  - **problems include (1) compatibility with full set of BGP features, (2) robustness, (3) scalability, (4) configuration/management interface**
- **Heuristics for analysis of RPSL**
- **Analysis of “backup routing”**

Work in Progress...

"An Analysis of BGP Convergence Properties"
Timothy G. Griffin, Gordon Wilfong

SIGCOMM'99

➡ **Model BGP, show static analysis is hard**

"Policy Disputes in Path Vector Protocols"
Timothy G. Griffin, F. Bruce Shepherd, Gordon Wilfong

ICNP '99

➡ **Define Stable Paths Problem and develop sufficient condition for "sanity"**

"A Safe Path Vector Protocol"
Timothy G. Griffin, Gordon Wilfong

INFOCOM'00

➡ **Dynamic solution based on histories**

"Stable Internet Routing without Global Coordination"
Lixin Gao, Jennifer Rexford

SIGMETRICS'00

➡ **Show that if certain guidelines are followed, then all is well.**

Links

www.research.att.com/~griffin

www.research.att.com/~griffin/interdomain.html