Control Plane Compression

Ryan Beckett

Aarti Gupta Ratul Mahajan David Walker

PRINCETON UNIVERSITY

INTENTIONET

South Africa: FNB solves crippling connectivity issues

July 25, 2016 • Finance, Southern Africa, Top Stories

Microsoft: misconfigured network device led to Azure outage

30 July 2012 | By Yevgeniy Sverdlik

BGP errors are to blame for Monday's Twitter outage, not DDoS attacks

No, your toaster didn't kill Twitter, an engineer did

Router Crashes Trigger Major Southwest IT System Failure

By: Chris Preimesberger | July 21, 2016

Unions want Southwest CEO removed after IT outage

BlackBerry outage could cost RIM $100 million

Massive route leak causes Internet slowdown

Posted by Andree Toonk – June 12, 2015 – BGP instability – No Comments

Xbox Live outage caused by network configuration problem

BY TODD BISHOP on April 15, 2013 at 9:27 am

Good news! Some Solutions

Data Plane Verification

Anteater	[Mai 2011]
HSA	[Kazemian 2012]
Veriflow	[Kurshid 2013]
NoD	[Lopes 2015]
Symmetries	[Plotkin 2016]
…	

Good news! Some Solutions

Data Plane Verification

Anteater	[Mai 2011]
HSA	[Kazemian 2012]
Veriflow	[Kurshid 2013]
NoD	[Lopes 2015]
Symmetries	[Plotkin 2016]
…	

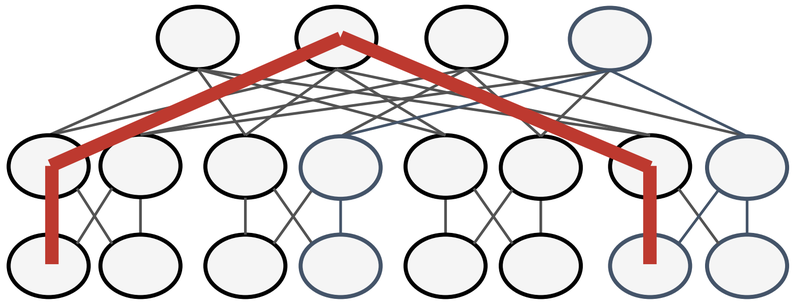
Control Plane Simulation

C-BGP	[Quotin 2005]
Batfish	[Fogel 2015]
…	

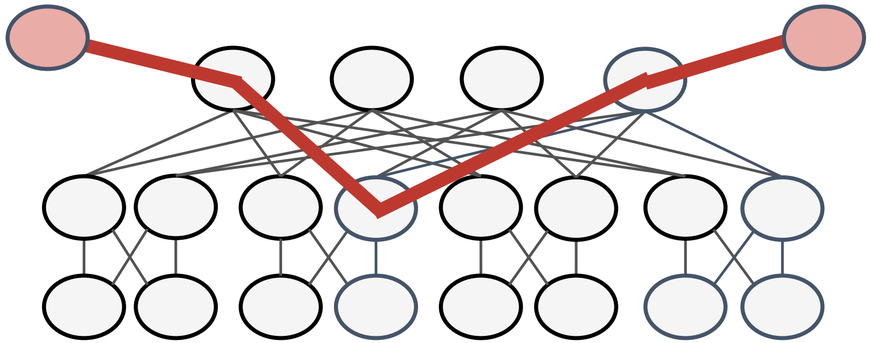
Control Plane Verification

Bagpipe	[Weitz 2016]
ARC	[Gember-Jacobsen 2016]
ERA	[Fayaz 2017]
MineSweeper	[Beckett 2017]
…	

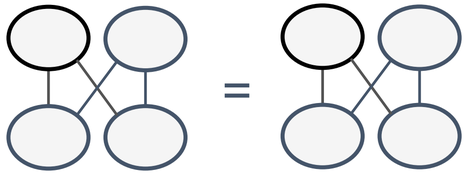
Properties



reachability



no transit

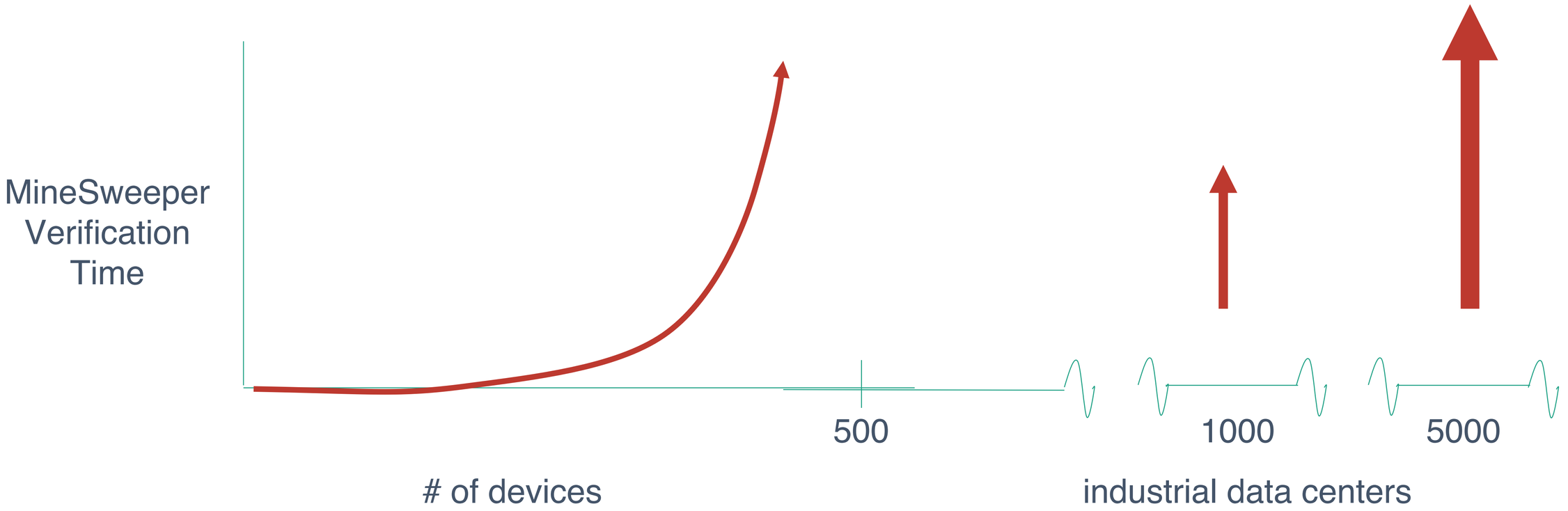


router or subnet equivalence

no loops

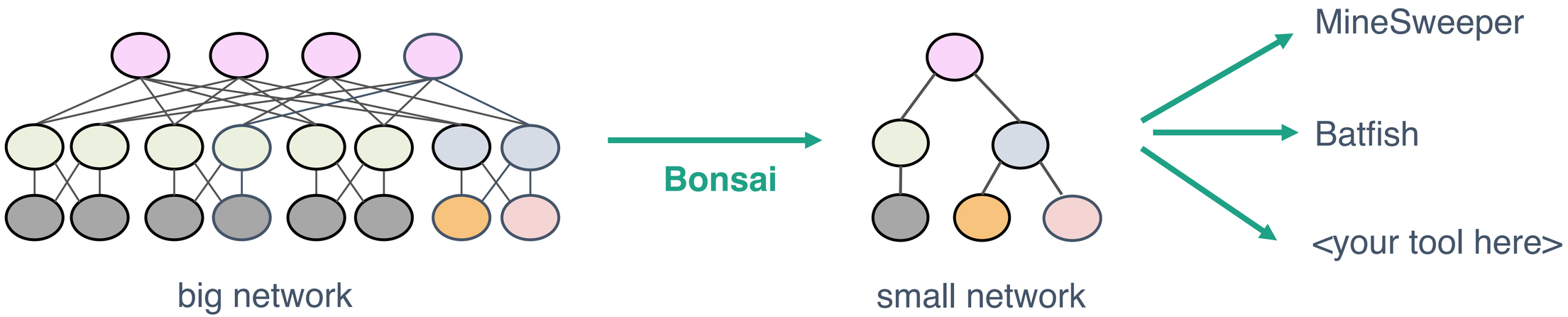
no black holes

A Problem of Scale



Other technologies, such as simulation, suffer similar, though less severe trends.

Our Contribution: Reduce the Scale

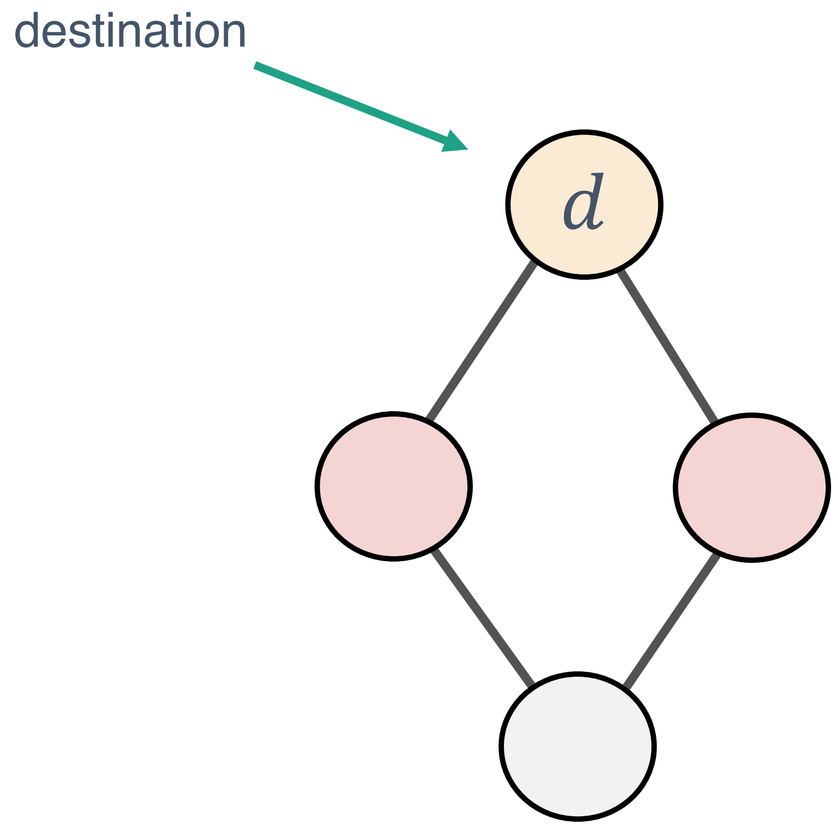


Empirical analysis: real networks reduced by 5-7x (# nodes); by 7-100x (# edges)

Theoretical analysis: we prove our algorithm generates behaviorally bisimilar networks

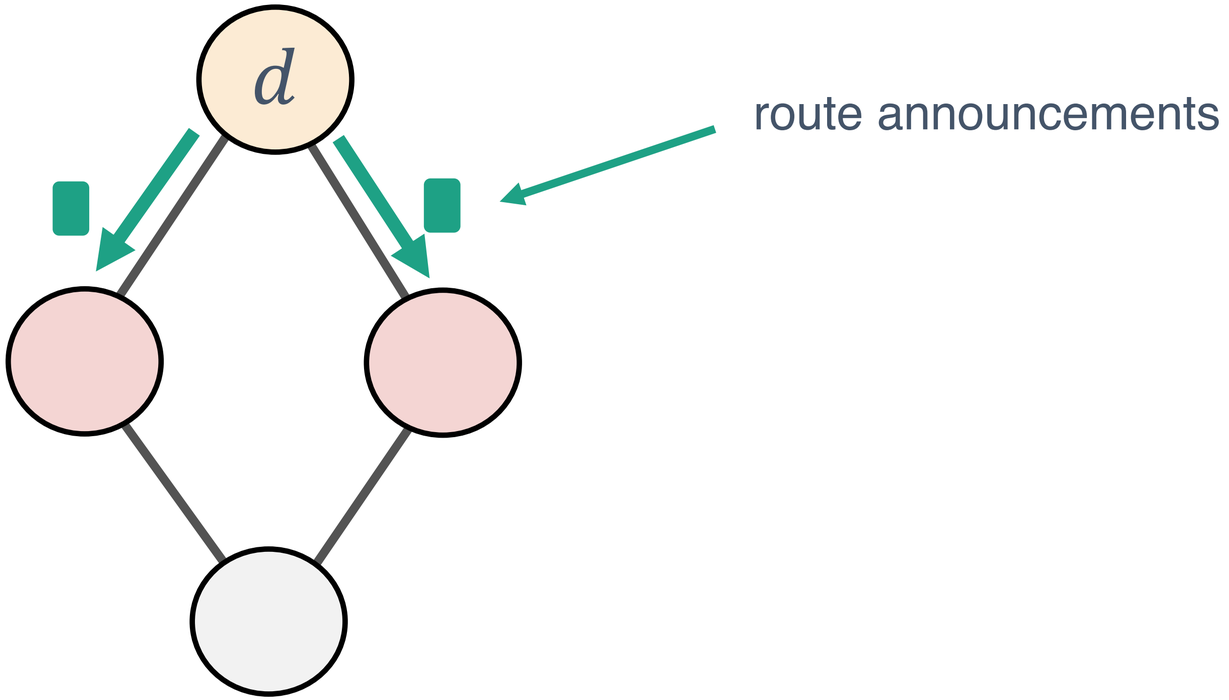
The Network Model

A Generic Routing Protocol

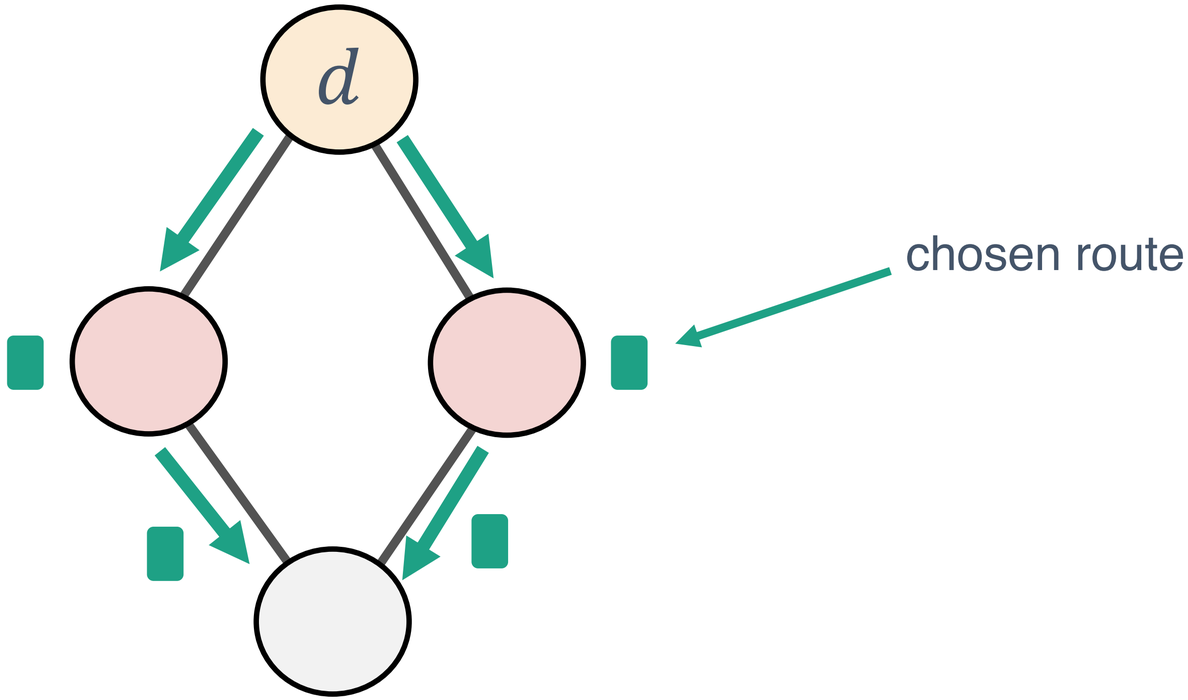


[Formal model builds on past work on stable paths [Griffin et al, 2002] or routing algebras [Sobrinho 2005] and work here at here at SIGCOMM 2018 by Daggitt *et al*.]

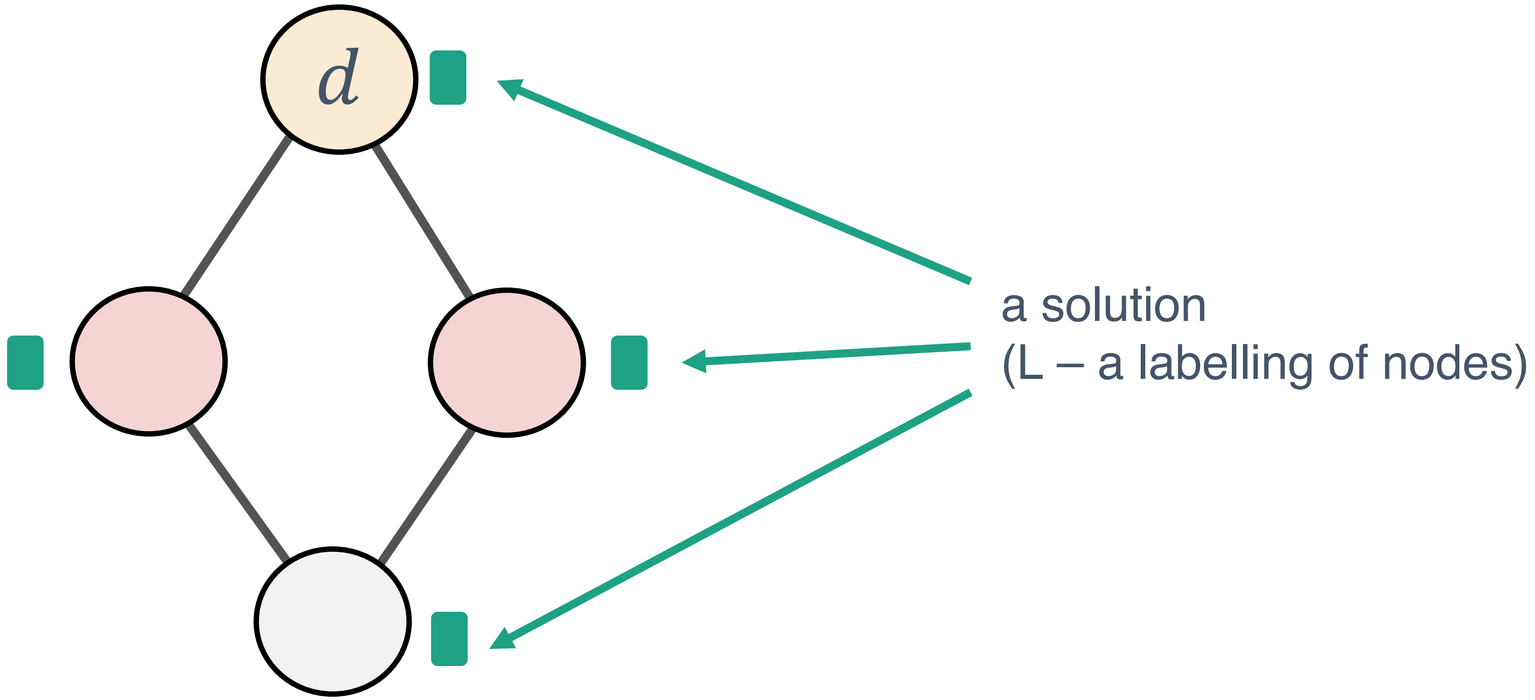
A Generic Routing Protocol



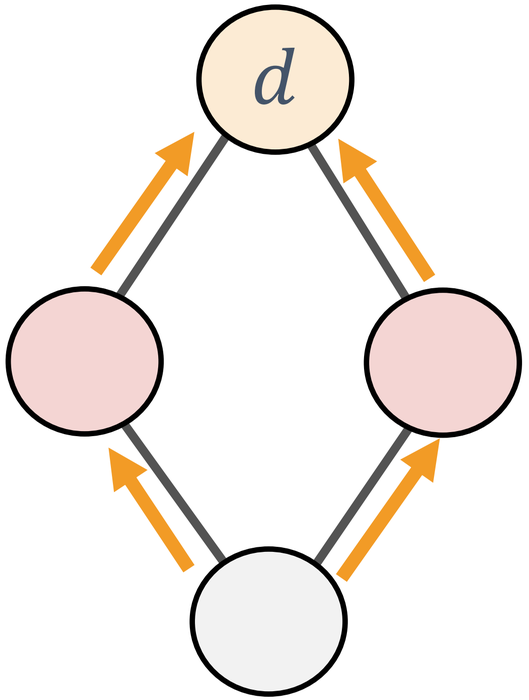
A Generic Routing Protocol



A Generic Routing Protocol



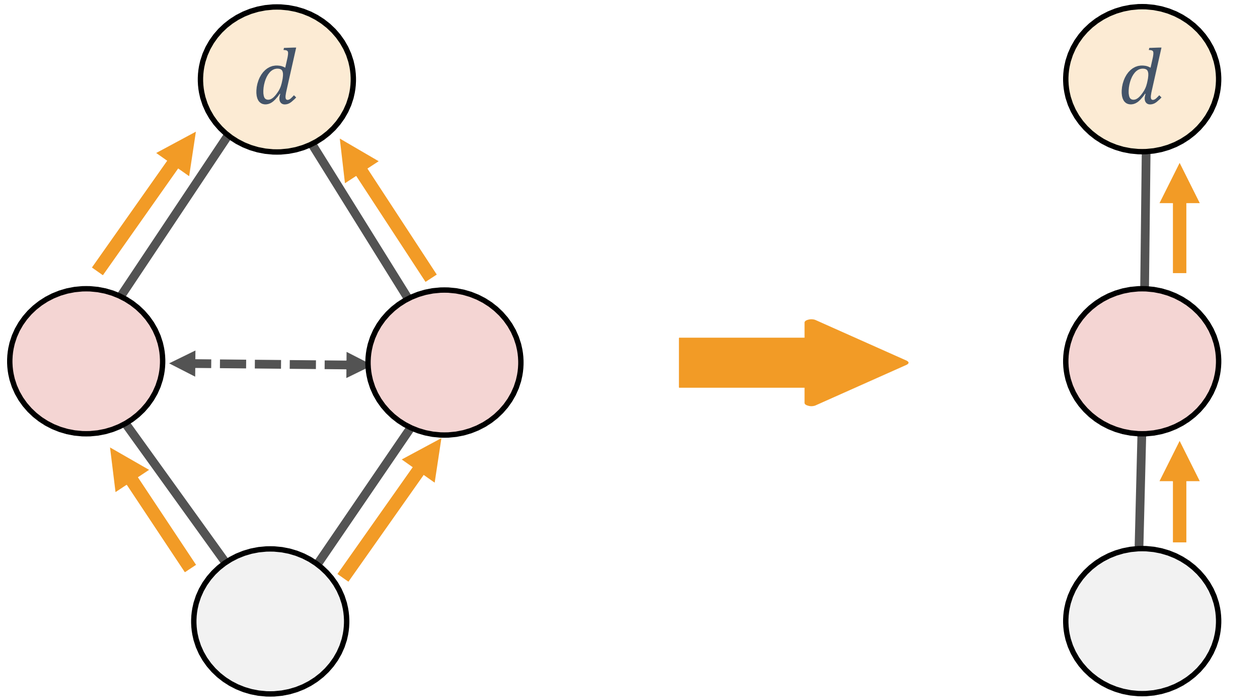
A Generic Routing Protocol



visual representation /
flow of traffic:

Valid Abstractions

Network Abstractions



Goal: Compute a small network with a “similar” solution to the big one

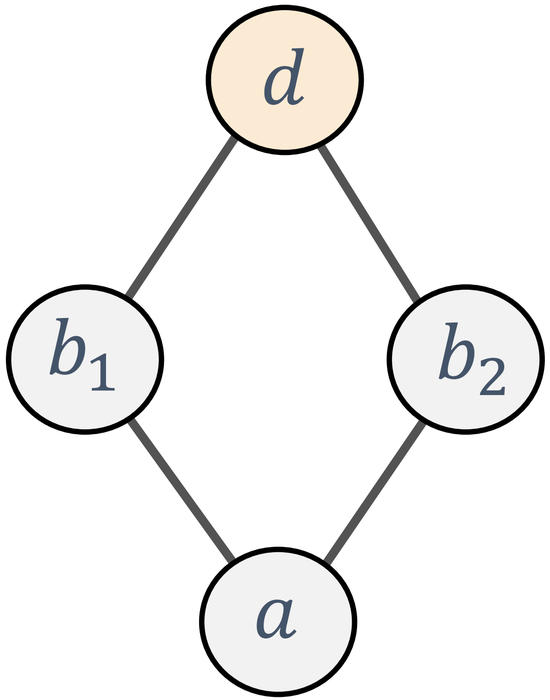
Constraint: *We can’t actually compute the solutions and compare them!*
We need a quick test that suffices to guarantee similarity.

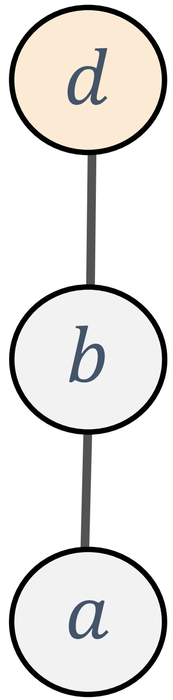
Network Abstractions

A pair of functions: (f, h)

abstracts route announcements

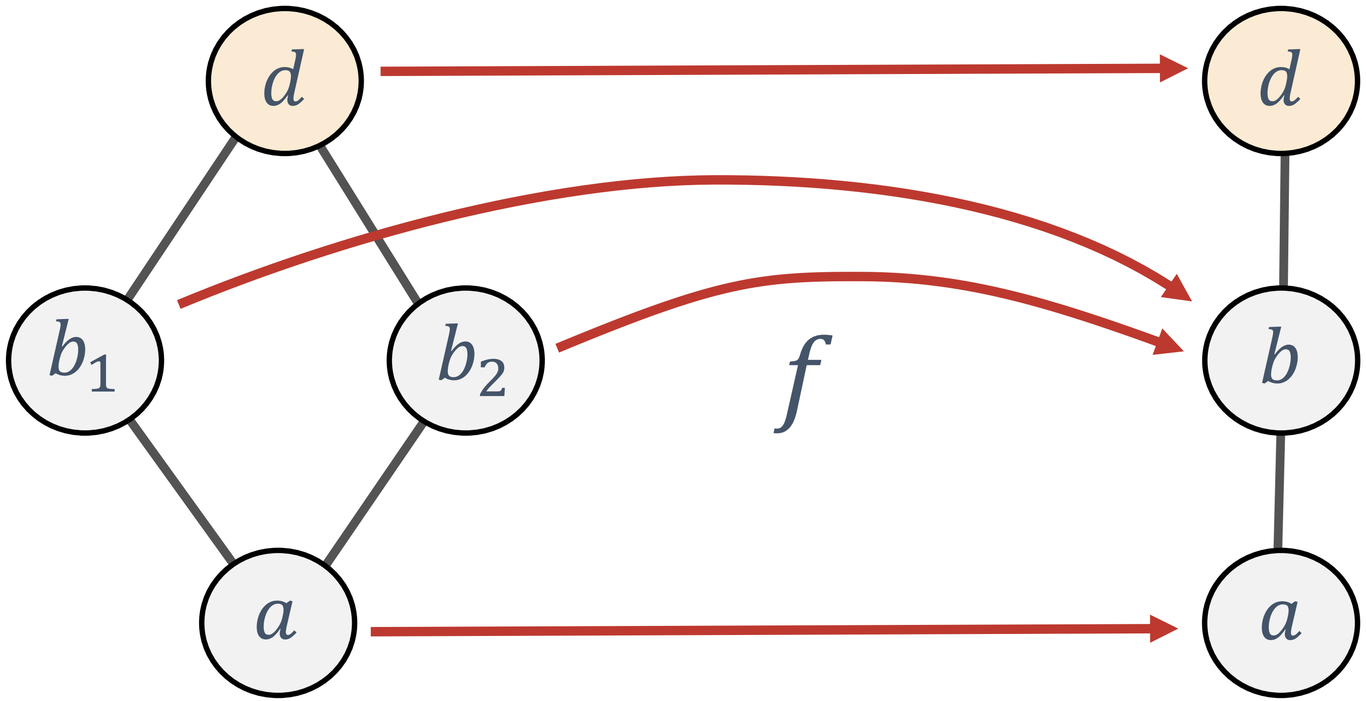
abstracts topology





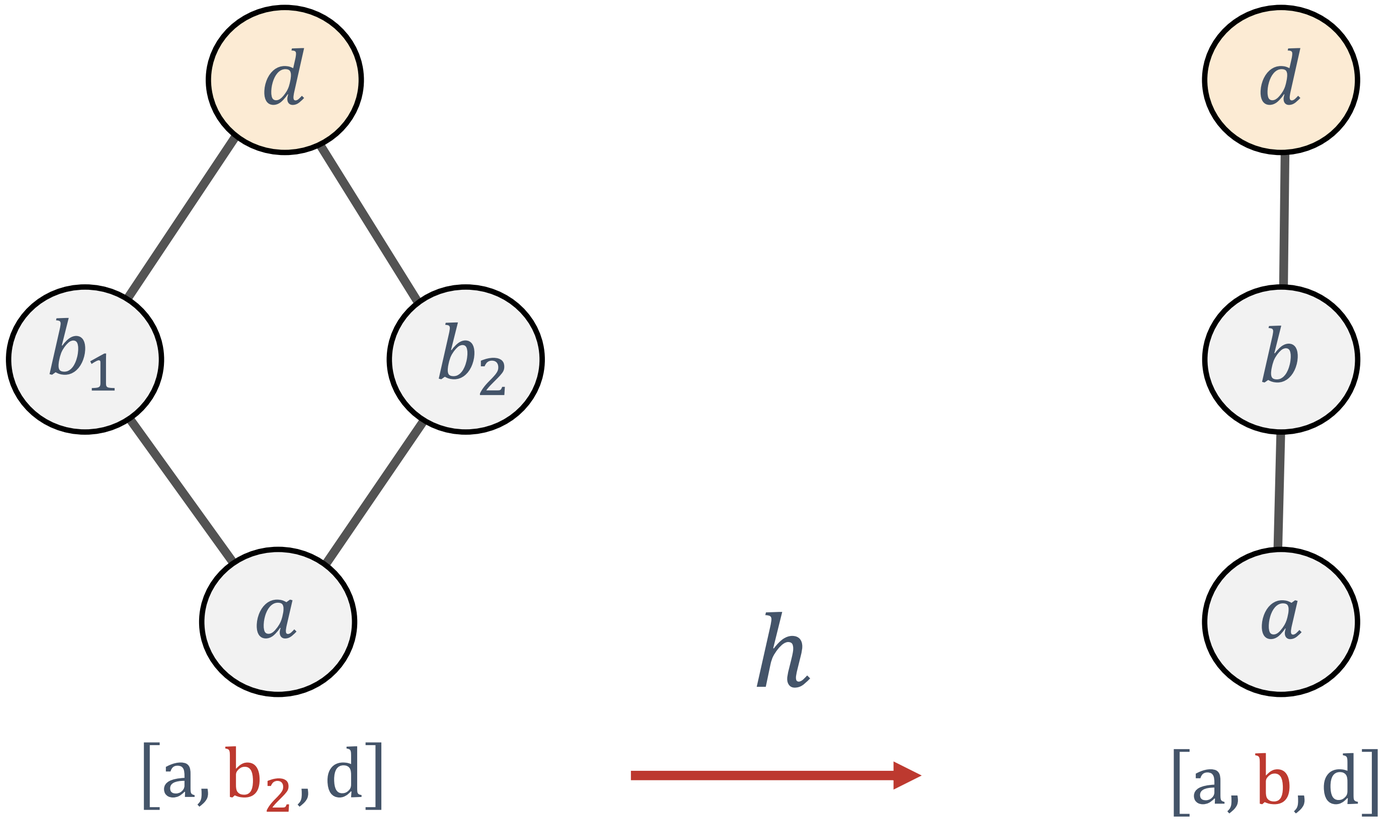
Network Abstractions

A pair of functions: (f, h)

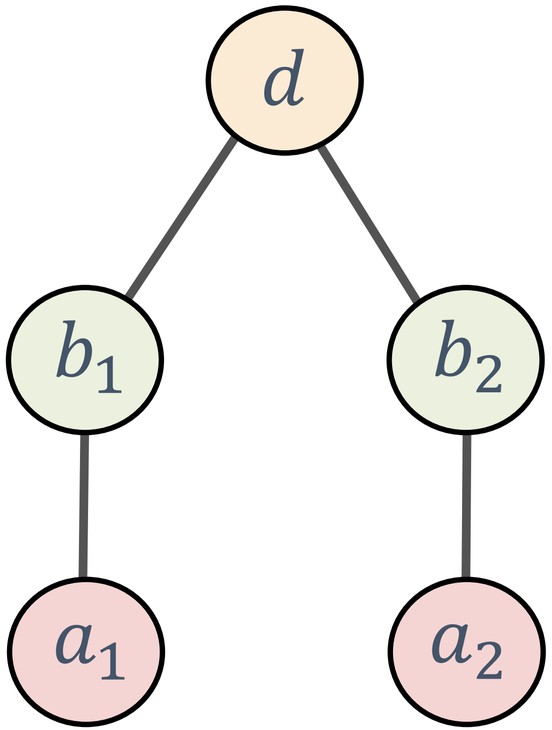


Network Abstractions

A pair of functions: (f, h)



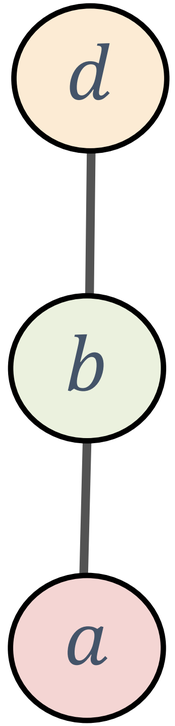
Abstraction Requirements



(1) Forall-exists requirement

concrete nodes must
have similar connections
as their abstract representatives

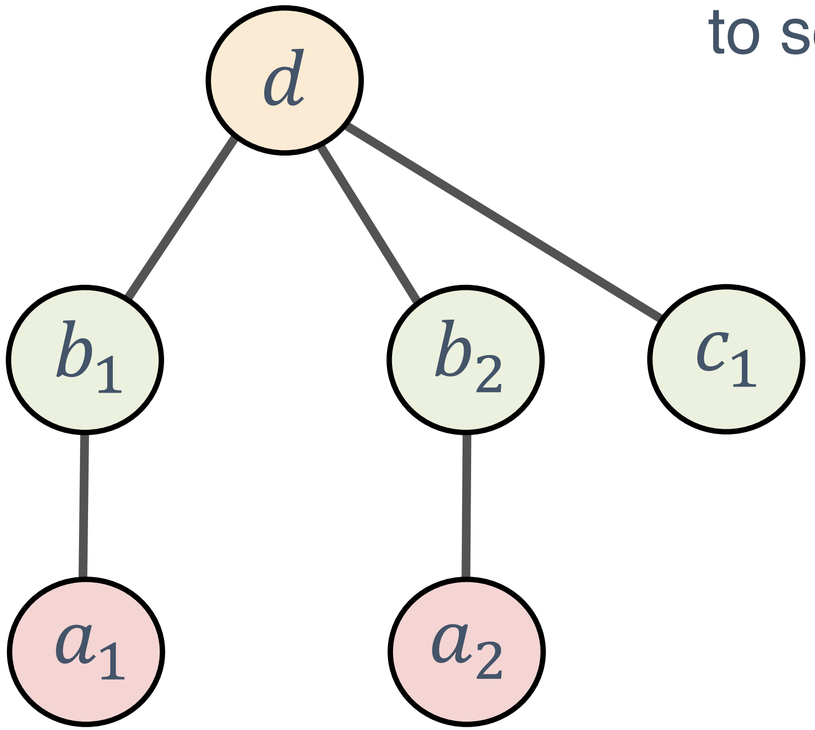
eg: Because the abstract green node has an edge to the abstract red node, ***all*** concrete green nodes must have an edge to ***some*** concrete red node



Abstraction Requirements

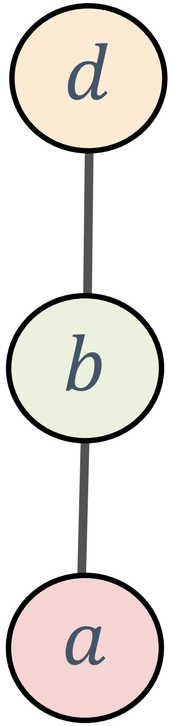
(1) Forall-exists requirement

All green nodes have an edge to some red node



Wrong.

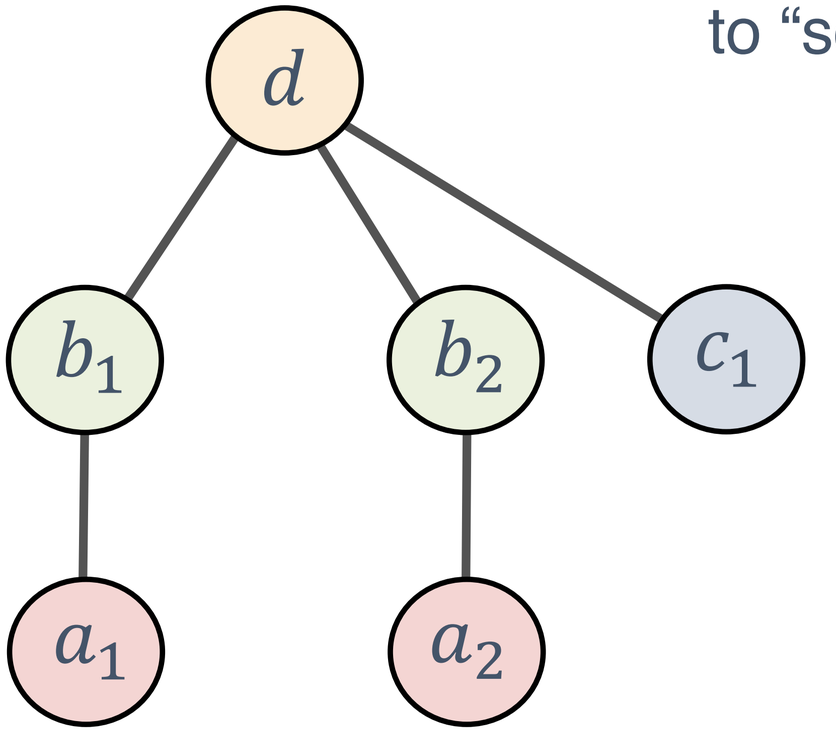
c_1 has no edge to a red node so it can't be green

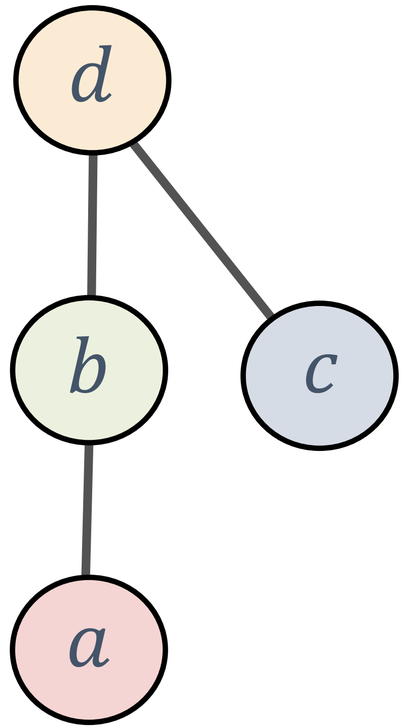


Abstraction Requirements

(1) Forall-exists requirement

Each green node has an edge to “some” red node

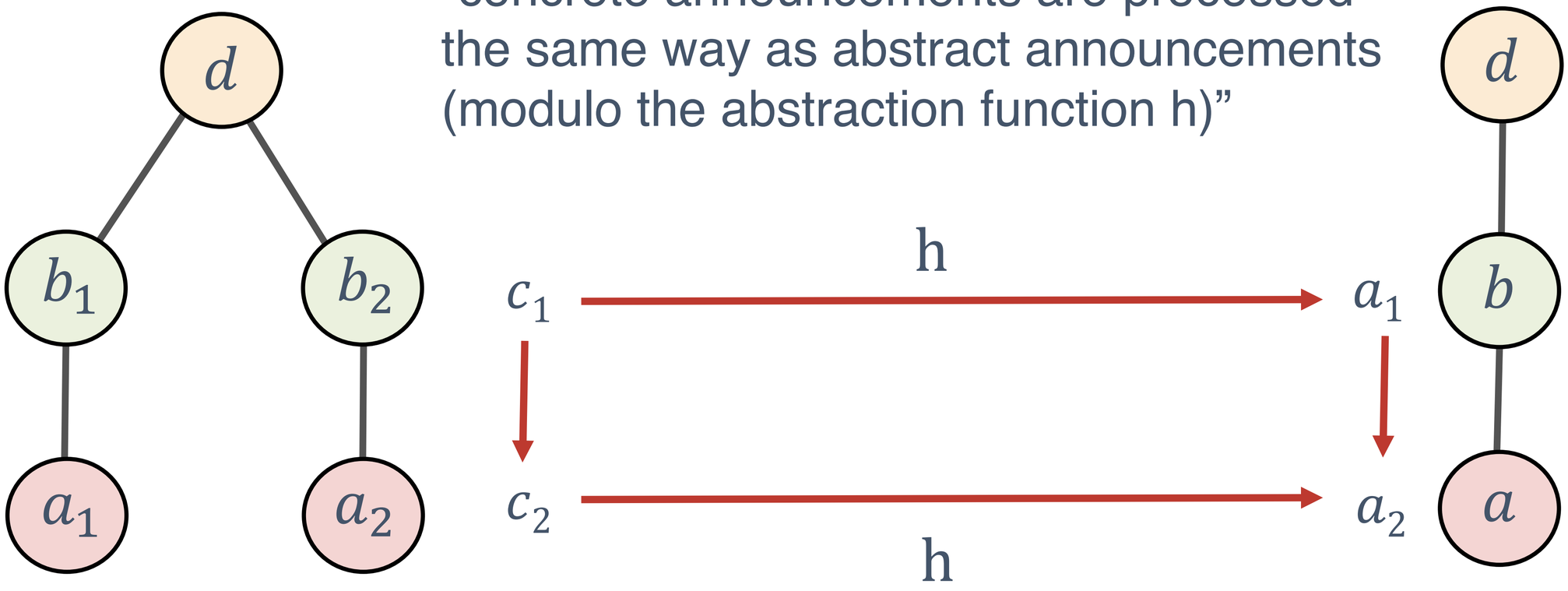




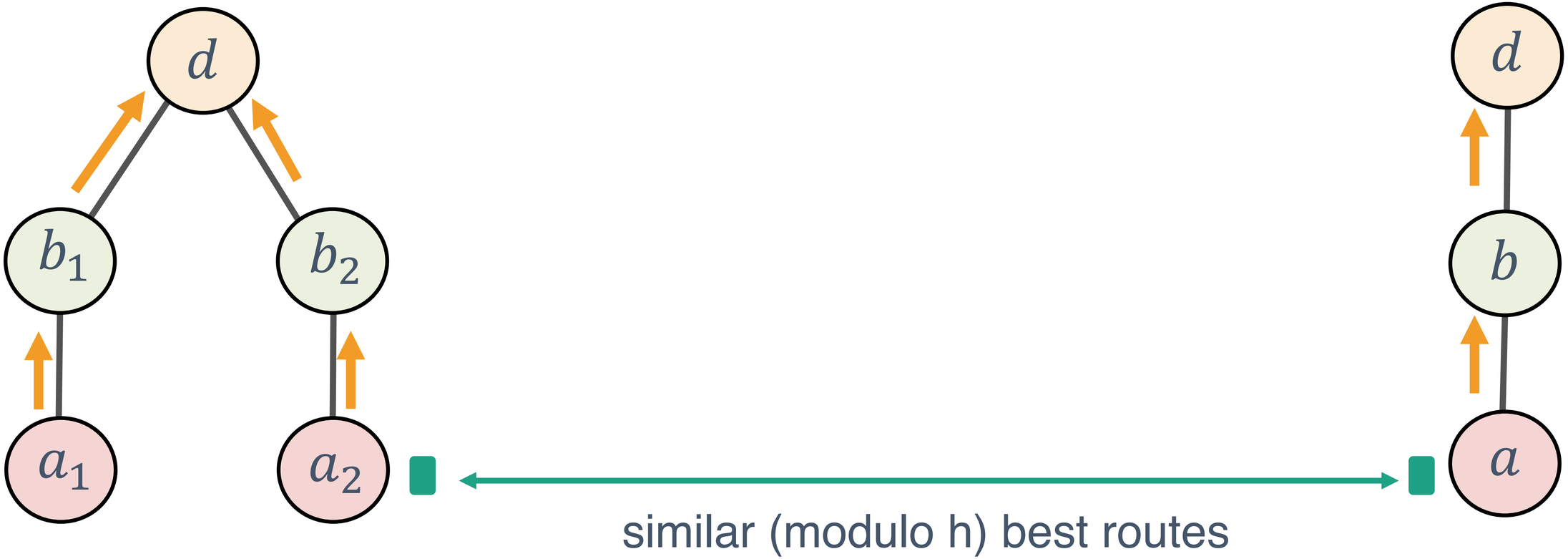
Abstraction Requirements

(2) Transfer-equivalence requirement

"concrete announcements are processed the same way as abstract announcements (modulo the abstraction function h)"



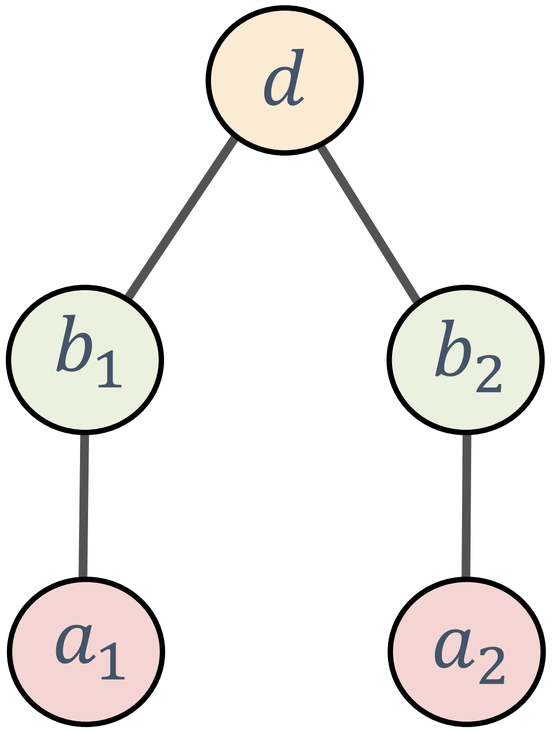
Theorem: If an **abstraction** satisfies the **forall-exists** requirement and the **transfer equivalence** requirement then it will compute **similar global solutions** as its related concrete network.

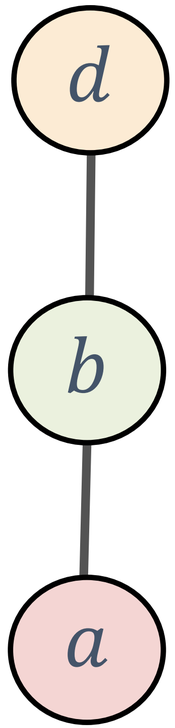


Corollary

Valid abstractions preserve:

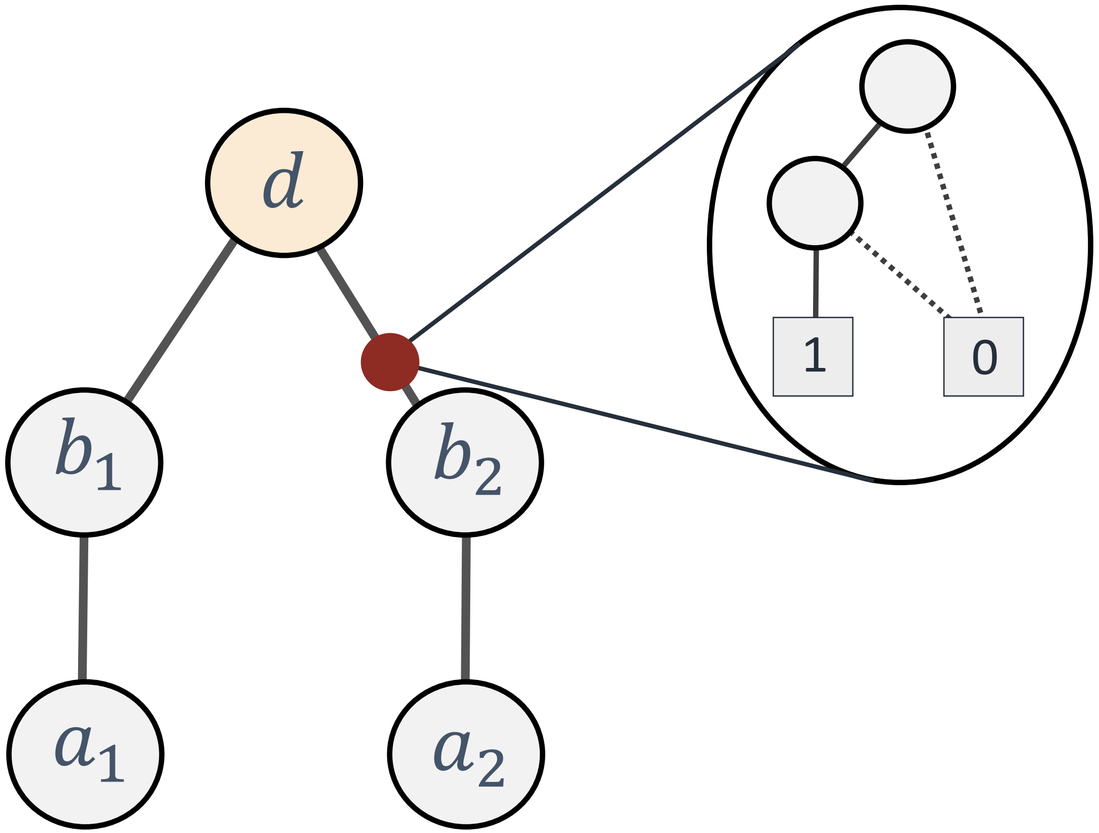
(1) Reachability
(2) Routing Loops
(3) Hop Count
(4) Multipath Consistency
(5) Waypointing





The algorithm:
How to find a valid abstraction

Checking for Transfer Equivalence



Binary Decision Diagrams (BDDs)

- Represents route-maps and ACLs
- Once BDDs have been built,
  we can test for transfer equivalence
  in constant time.

Finding an Abstraction: The Algorithm

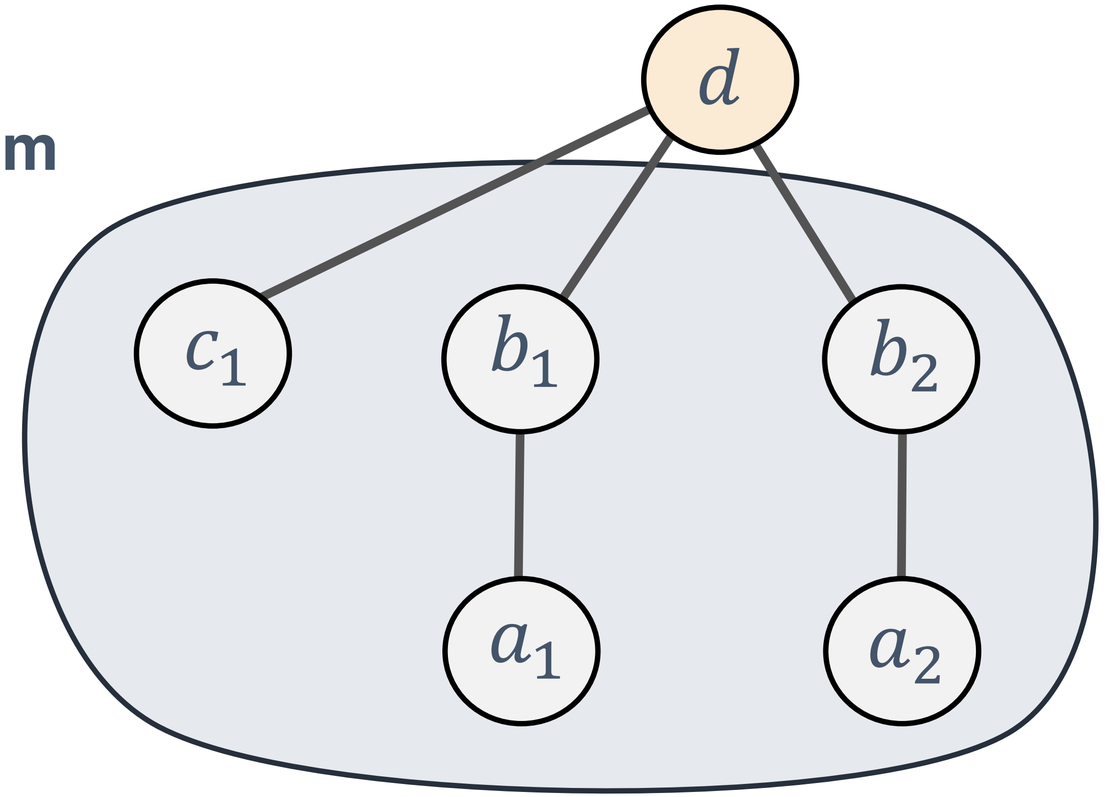
Greedy Abstraction Refinement Algorithm

- Start with 2 abstract nodes
- Repeatedly split until a valid abstraction is found.

Finding an Abstraction: The Algorithm

Greedy Abstraction Refinement Algorithm

- Start with 2 abstract nodes
- Repeatedly split until a valid abstraction is found.

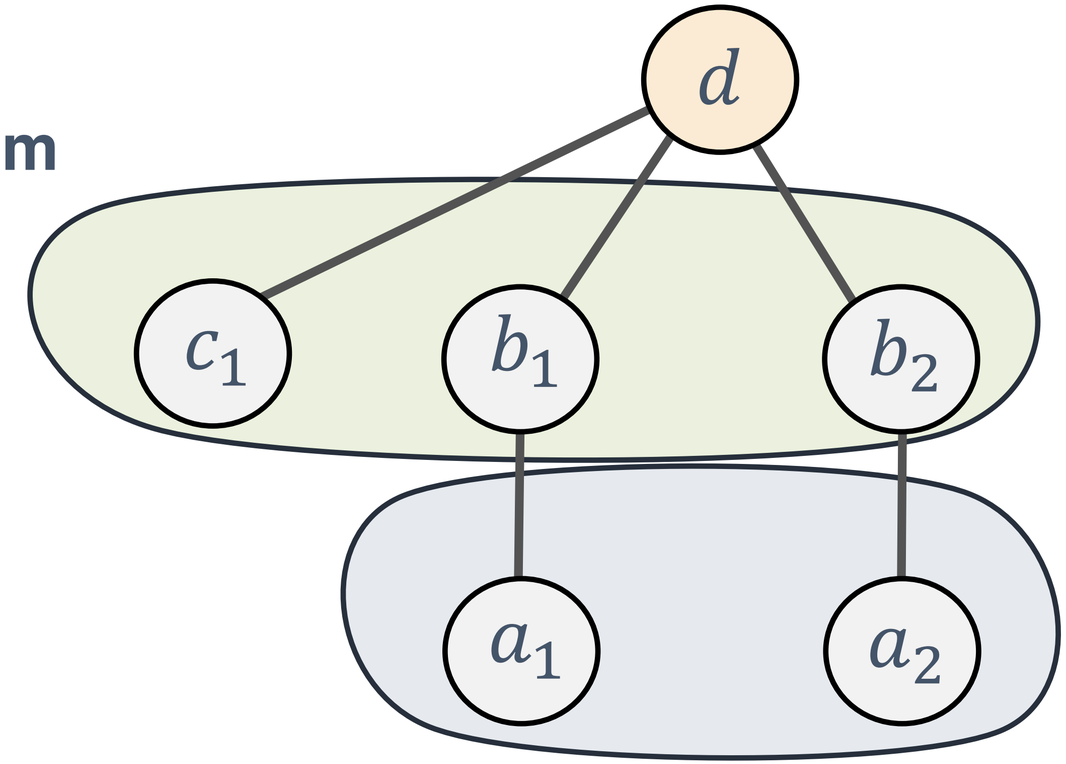


topological forall-exists condition is violated:
b_1 has an edge to orange node, but a_1 does not.

Finding an Abstraction: The Algorithm

Greedy Abstraction Refinement Algorithm

- Start with 2 abstract nodes
- Repeatedly split until a valid abstraction is found.

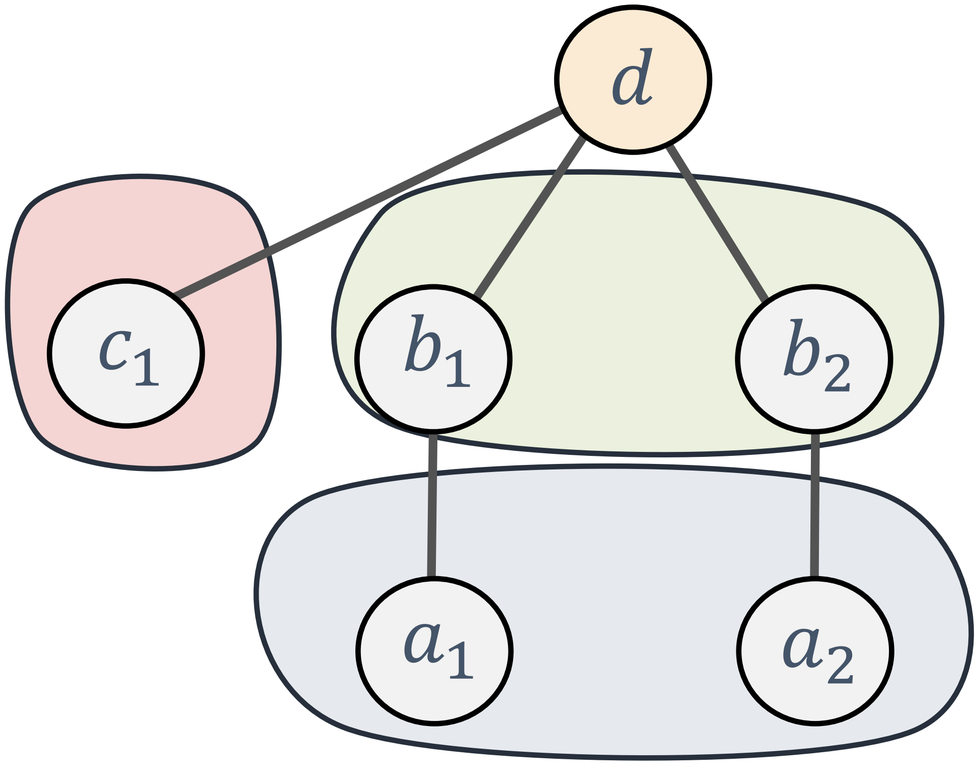


topological forall-exists condition is violated:
b_1 has an edge to a blue node, but c_1 does not

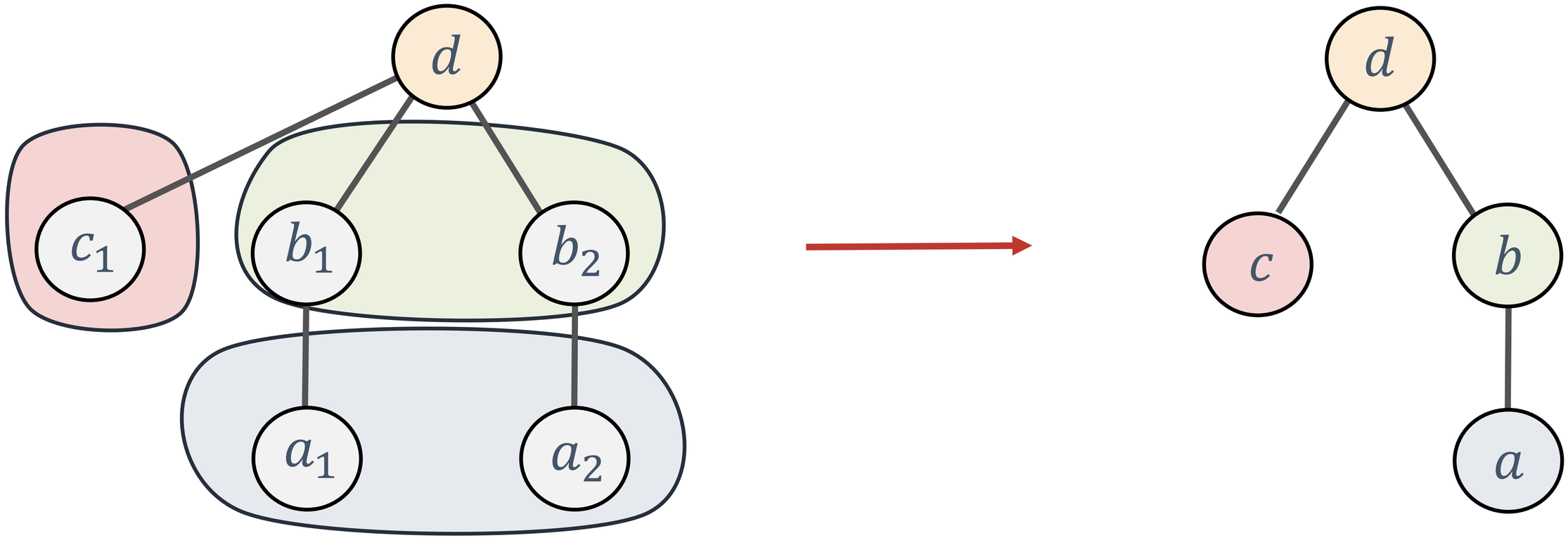
Finding an Abstraction: The Algorithm

Greedy Abstraction Refinement Algorithm

- Start with 2 abstract nodes
- Repeatedly split until a valid abstraction is found.



Finding an Abstraction: The Algorithm



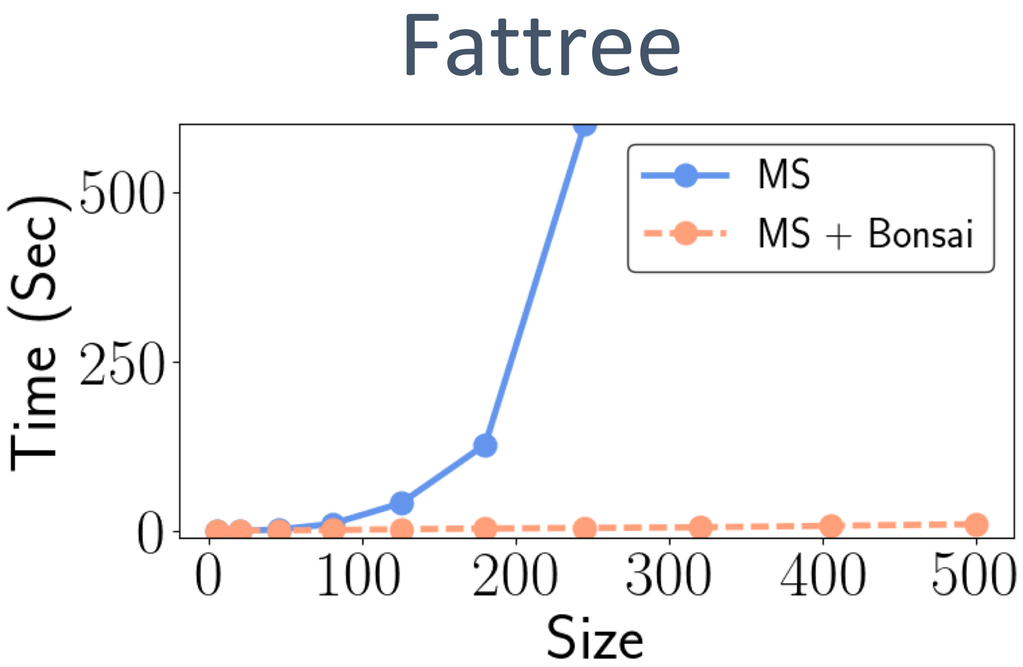
An Aside: BGP Behaving Badly

- *You might think* that if 2 BGP nodes have syntactically identical configurations then they process all routes the same way (ie, transfer equivalence holds)
- Spoiler: *They might not! BGP loop detection discards routes differently.*
- Comment: If transfer equivalence doesn't hold, the algorithm fails to compress
- Fun fact: We prove a node can have up to k+1 different behaviors, where k is the # of different local preferences used.
- See the paper (and Ryan's thesis) for proofs and a revised algorithm for BGP.

Evaluation

Synthetic Benchmarks

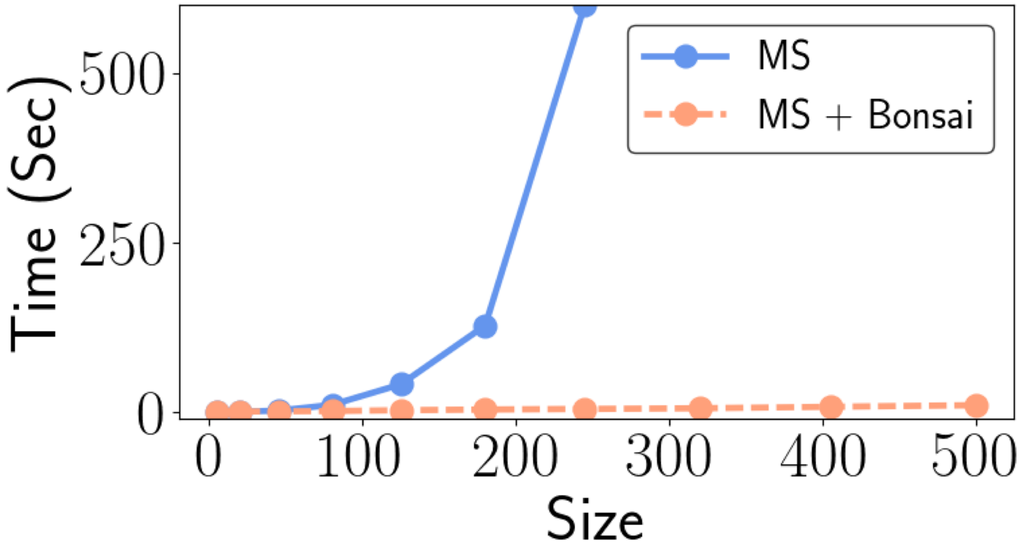
[MineSweeper verifying all-pairs reachability with shortest paths policy]



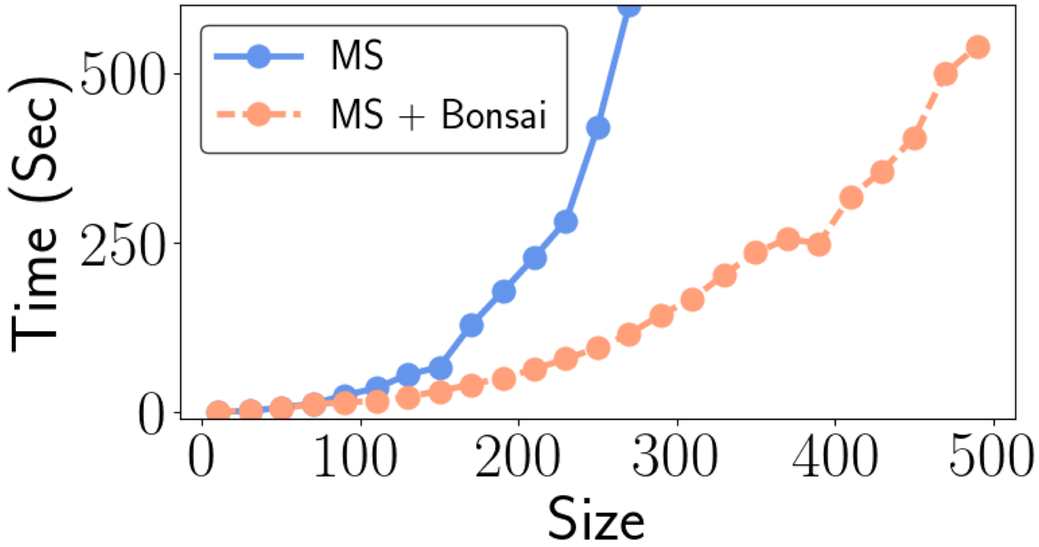
Synthetic Benchmarks

[MineSweeper verifying all-pairs reachability with shortest paths policy]

Fattree



Ring



Case Studies

Datacenter

- Roughly **600,000 lines of configuration** for **197 devices**
- Only **26 unique “roles”**
- Compression takes **~15.5 seconds** per destination class (DC) for 1269 DCs
- Number of nodes compressed on average by **6.6x** and edges by **112x**

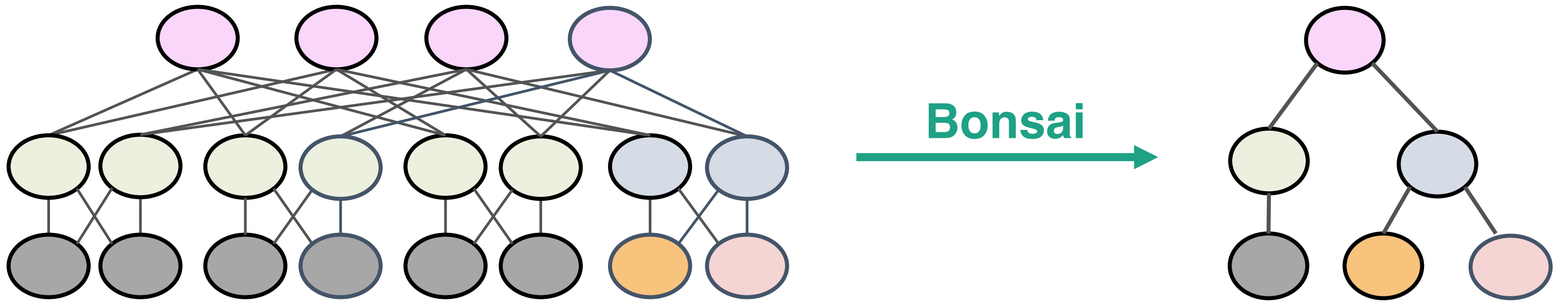
WAN

- Roughly **600,000 lines of configuration** for **1086 devices**
- Only **137 unique “roles”**
- Compression takes **~1.8 seconds** per DC for 845 DCs
- Number of nodes compressed by **5.2x** and edges by **7.2x**
- Note: MineSweeper still doesn’t scale due to the protocols used; Batfish does

Bonsai Limitations

- Not guaranteed to find the **optimal** abstraction (though often good)
- Properties can not depend on the **number of edges/neighbors/paths**
  - Fault tolerance properties are not preserved
- Whether or not Bonsai preserves **divergence** is an open question

Summary: Control Plane Compression



- The Bonsai algorithm finds compresses real networks by a factor of 5-7 in the number of nodes and 5-100 in the number of edges.
- It preserves many path properties, such as reachability, but not fault tolerance.
- We have proven it correct with respect to a generic routing protocol.