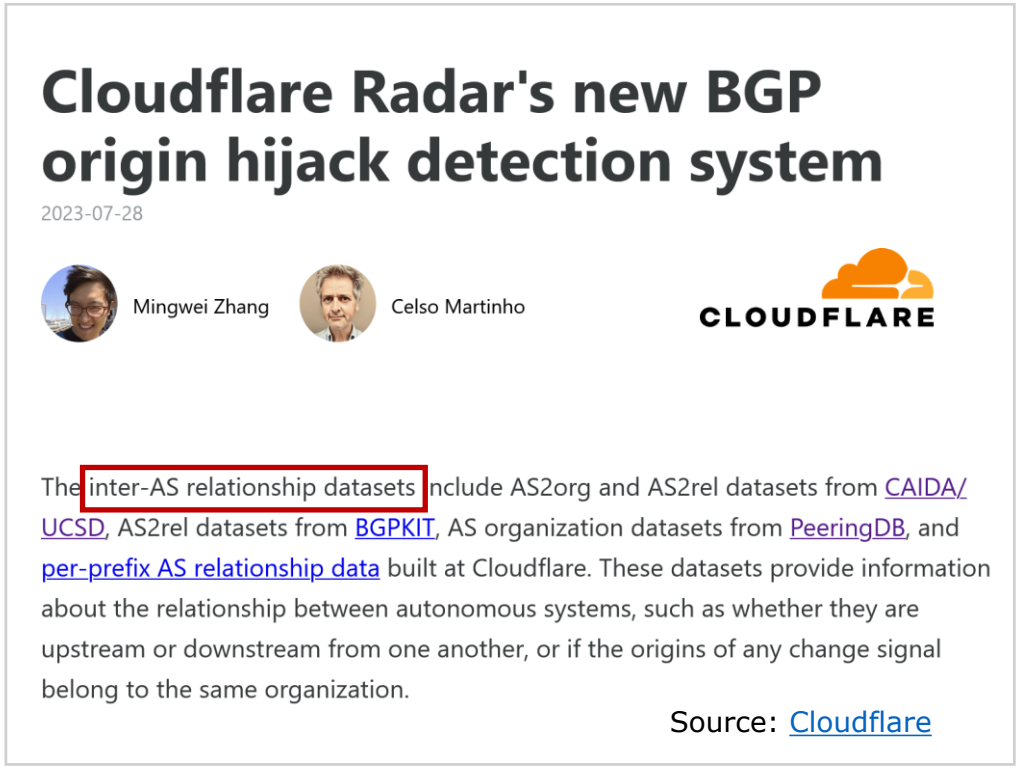
Adversarial Manipulation Risks in Internet Measurement: A Case Study of ASRank

Speaker: Yihao Chen (Tsinghua University)

In collaboration with: Yi Xu, Ke Xu, Qi Li, Jianping Wu

AS Business Relationship

- AS relationships largely determine BGP routing policies
- Knowledge of them is important for many downstream tasks



Cloudflare Radar's new BGP origin hijack detection system

2023-07-28

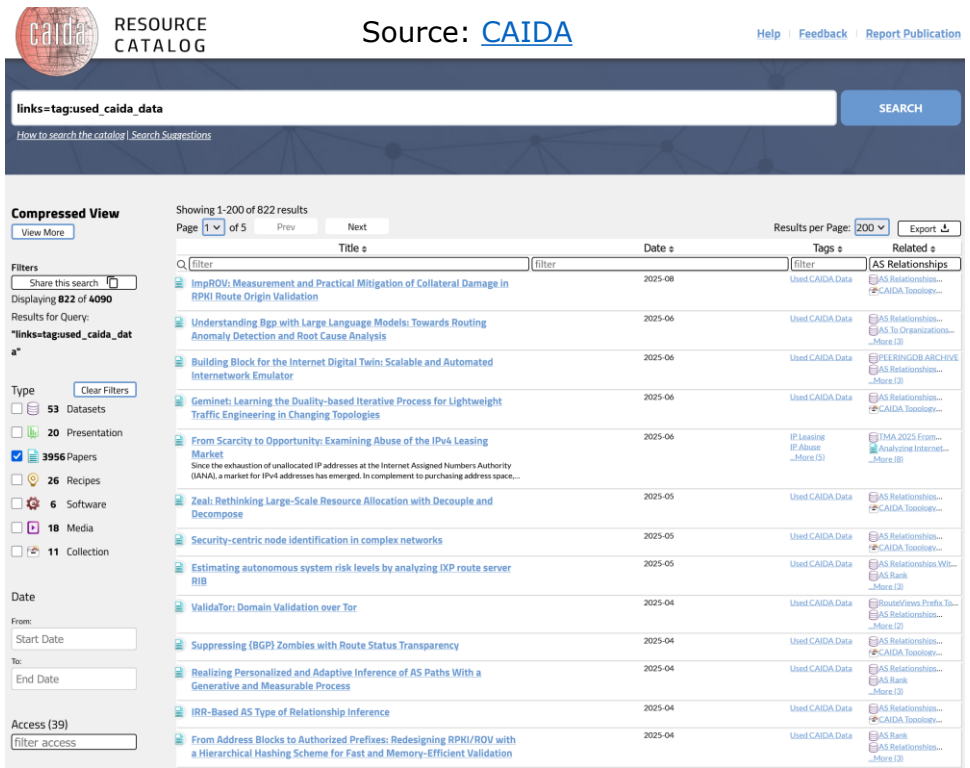
Mingwei Zhang Celso Martinho

CLOUDFLARE

The inter-AS relationship datasets include AS2org and AS2rel datasets from CAIDA/UCSD, AS2rel datasets from BGPKIT, AS organization datasets from PeeringDB, and per-prefix AS relationship data built at Cloudflare. These datasets provide information about the relationship between autonomous systems, such as whether they are upstream or downstream from one another, or if the origins of any change signal belong to the same organization.

Source: Cloudflare

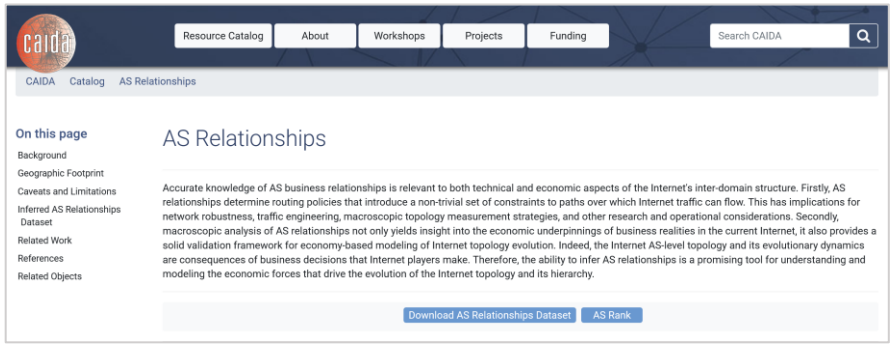
Cloudflare Radar uses AS relationship datasets



>800 studies cite CAIDA's AS relationship datasets

ASRank's Foundational Role

- AS relationship knowledge relies on algorithmic inference from empirical routing data
- CAIDA's ASRank algorithm is arguably the most influential and widely used



CAIDA AS relationship datasets

generated by

AS Relationships, Customer Cones, and Validation

Matthew Luckie
CAIDA / UC San Diego
mjl@caida.org

Bradley Huffaker
CAIDA / UC San Diego
bradley@caida.org

Amogh Dhamdhere
CAIDA / UC San Diego
amogh@caida.org

Vasileios Giotsas
University College London
V.Giotsas@cs.ucl.ac.uk

kc claffy
CAIDA / UC San Diego
kc@caida.org

Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

ASRank (IMC'13)

dependent on

UNARI (CoNEXT'19) **ProbLink** (NSDI'19)
TopoScope (IMC'20) **RouteInfer** (PAM'22)
……

state-of-the-art inference methods

ASRank's Foundational Role

- AS relationship knowledge relies on algorithmic inference from empirical routing data
- CAIDA's ASRank algorithm is arguably the most influential and widely used

Does ASRank remain reliable under adversarial manipulation?

- CAIDA's inference pipeline is transparent
- No prior work has examined the risk

AS Relationships, Customer Cones, and Validation

Matthew Luckie
CAIDA / UC San Diego
mjl@caida.org

Bradley Huffaker
CAIDA / UC San Diego
bradley@caida.org

Amogh Dhamdhere
CAIDA / UC San Diego
amogh@caida.org

Vasileios Giotsas
University College London
V.Giotsas@cs.ucl.ac.uk

kc claffy
CAIDA / UC San Diego
kc@caida.org

Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes

1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

ASRank (IMC'13)

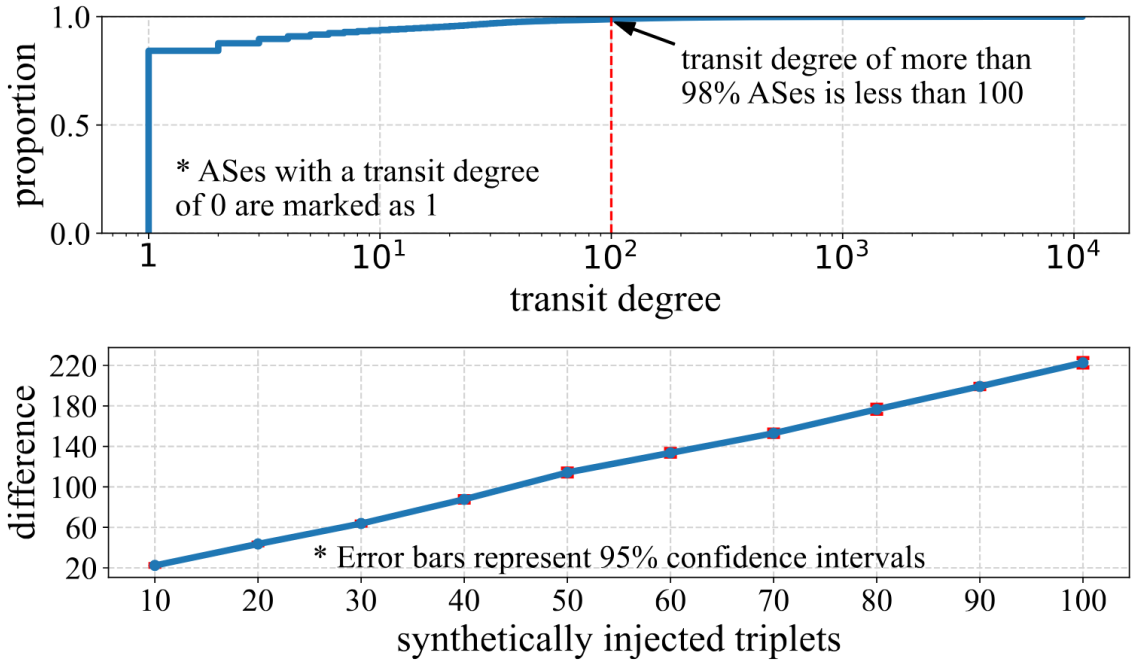
ASRank is Fragile

- Its main step is order-dependent and greedy based on empirical AS triplets
- Subtle perturbation of the order can lead to cascading misclassification

Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes

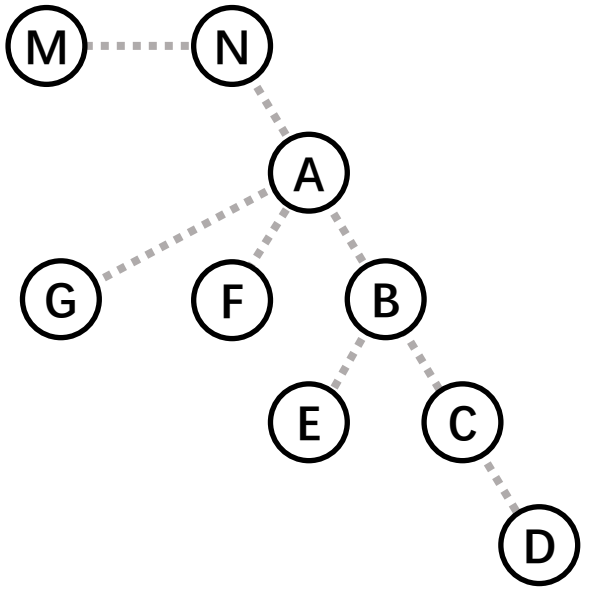
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)



A Minimal Example

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes



Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

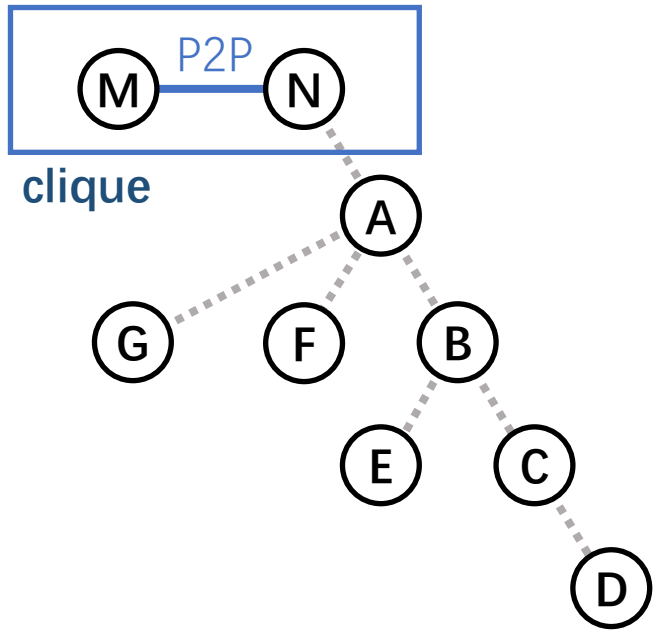
A Minimal Example (Step 3: infer P2P among clique)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

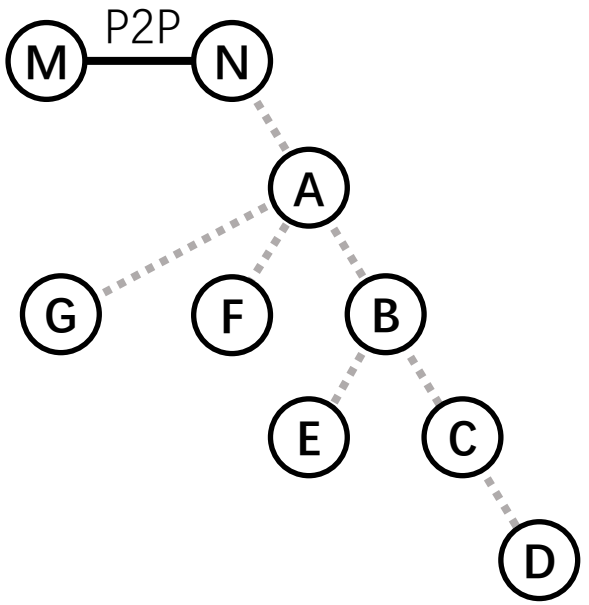


Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes



Resolved links

Algorithm 1 AS relationship inference algorithm.

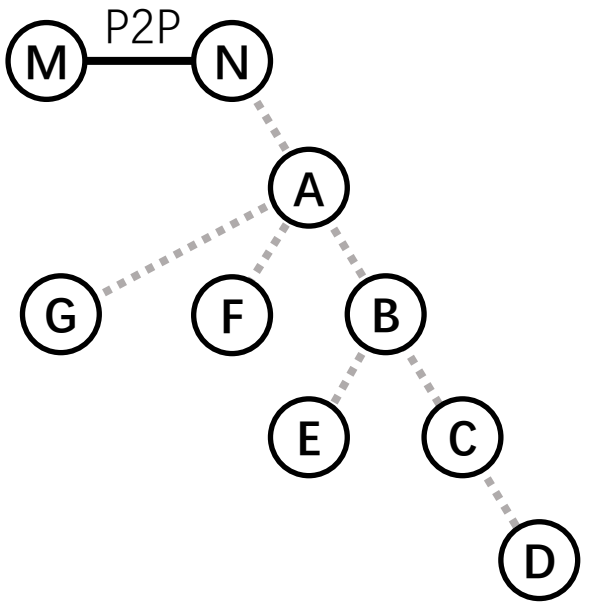
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

1. Rank ASes by transit degree



Resolved links

Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

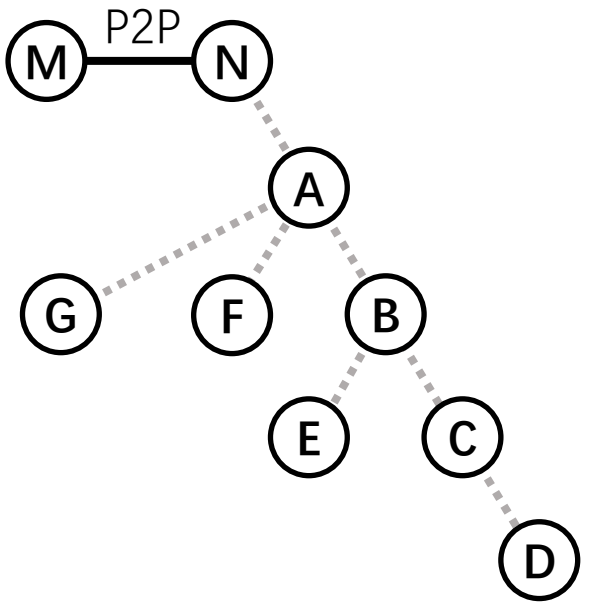
Empirical input routes

Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's distinct neighbors in AS triplets where the subject AS is in the middle position.



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*

path 2: ... *M N A B E*

path 3: ... *M N A F*

path 4: ... *M N A G*

Empirical input routes

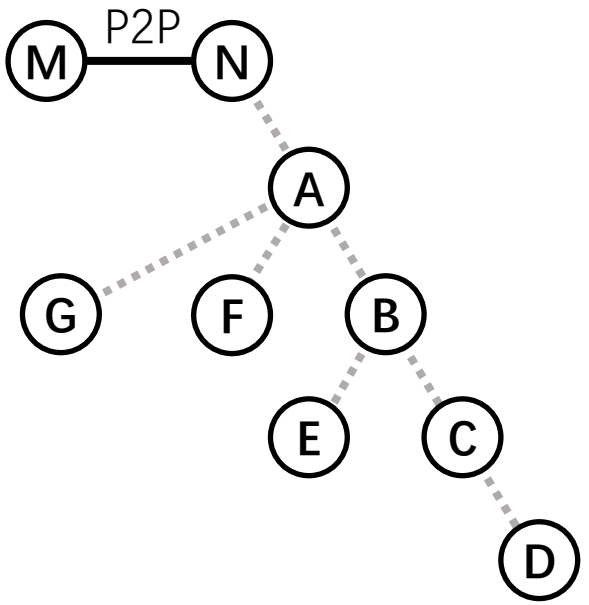
Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes

1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's distinct neighbors in AS triplets where the subject AS is in the middle position.



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: … *M N A B C D*

path 2: … *M N A B E*

path 3: … *M N A F*

path 4: … *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

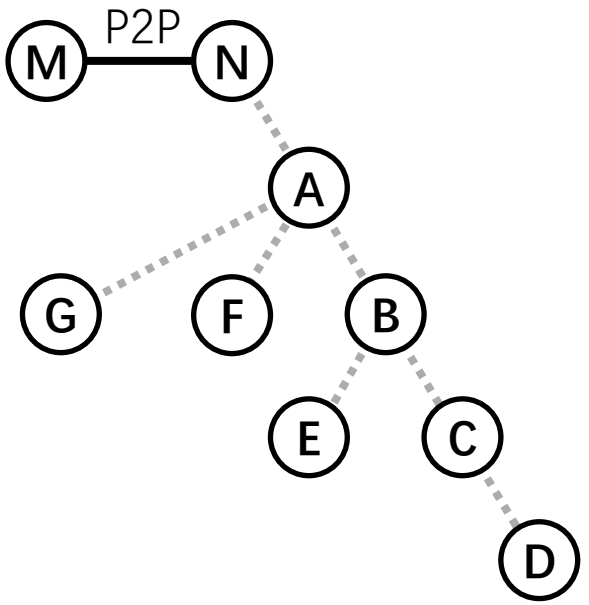
Require: AS paths, Allocated ASNs, IXP ASes

1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's distinct neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

Empirical input routes

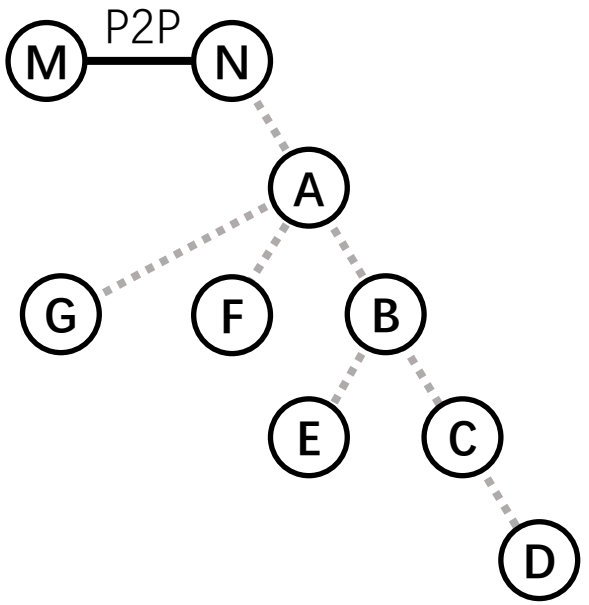
Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's distinct neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4
- TD(B) = 3



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

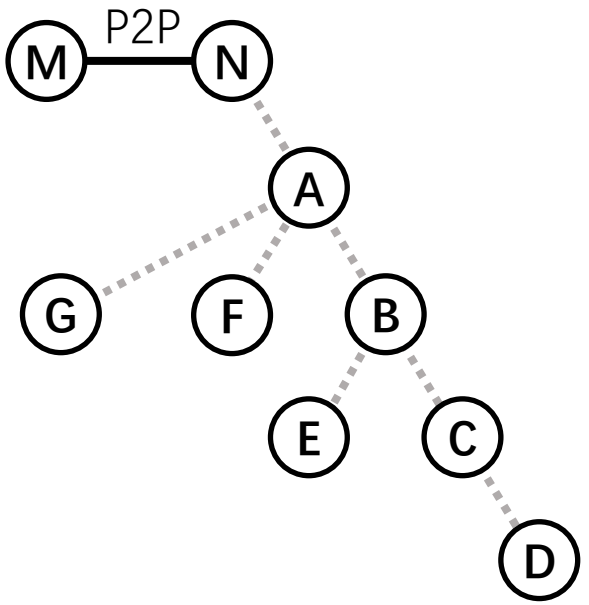
Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's distinct neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4
- TD(B) = 3
- TD(C) = 2



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

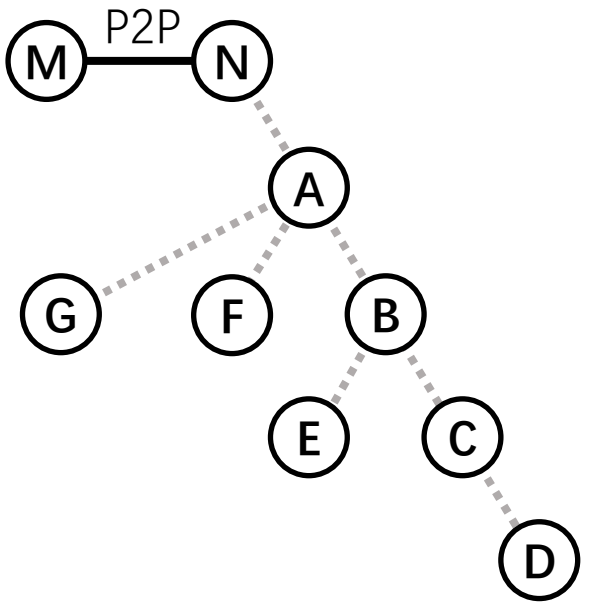
Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's distinct neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4
- TD(B) = 3
- TD(C) = 2
- TD(D)=TD(E)=TD(F)=TD(G)=0



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

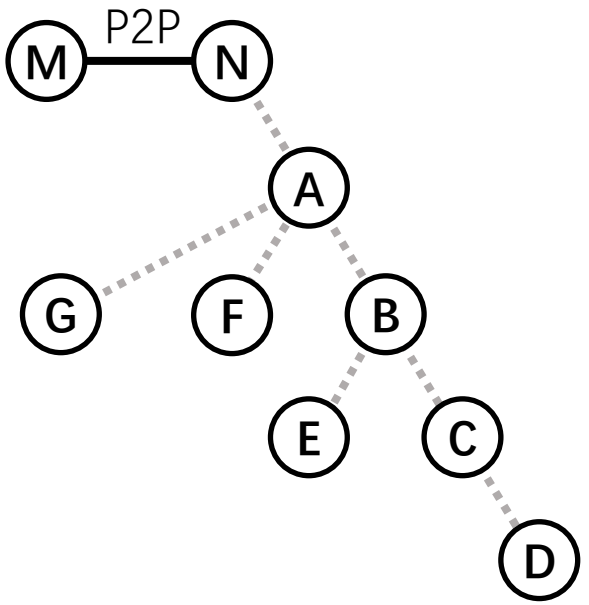
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's distinct neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4
- TD(B) = 3
- TD(C) = 2
- TD(D)=TD(E)=TD(F)=TD(G)=0

Top-down order: A,B,C,D,E,F,G



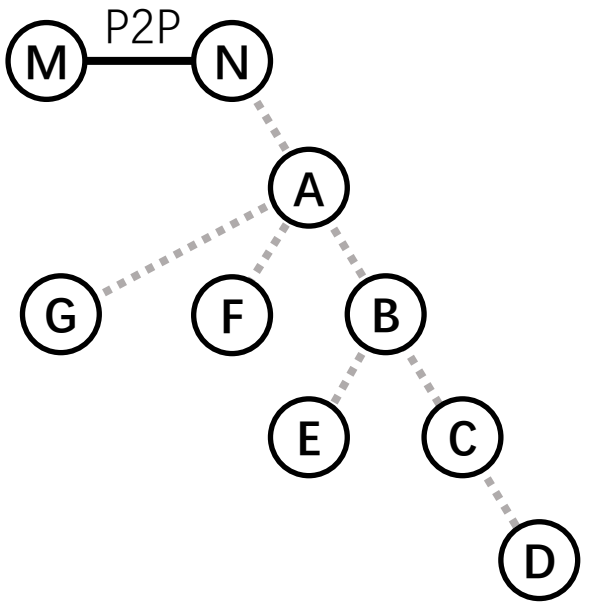
Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G



Resolved links

Algorithm 1 AS relationship inference algorithm.

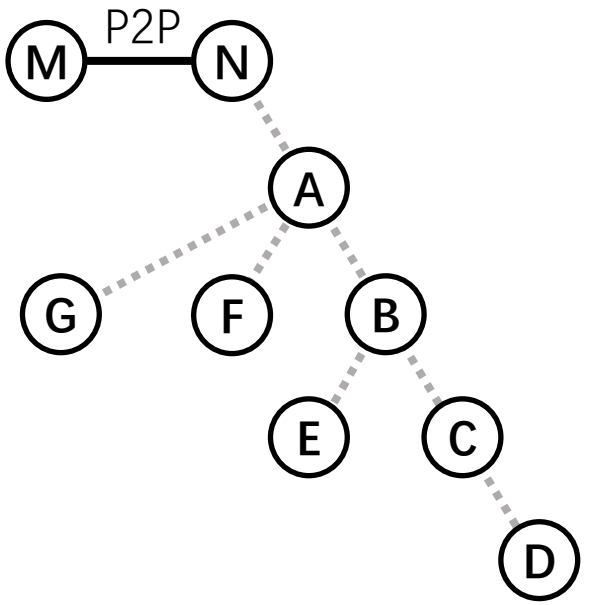
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G
2. Infer P2C heuristically



Resolved links

Algorithm 1 AS relationship inference algorithm.

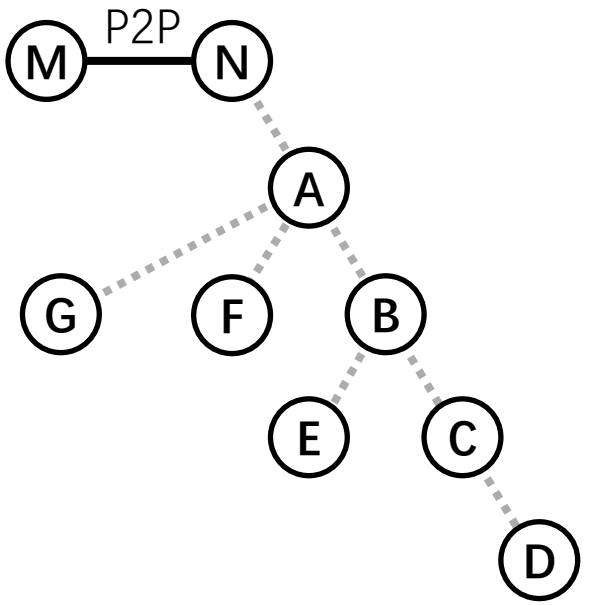
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G
2. Infer P2C heuristically
Visit each AS in top-down order.



Resolved links

Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

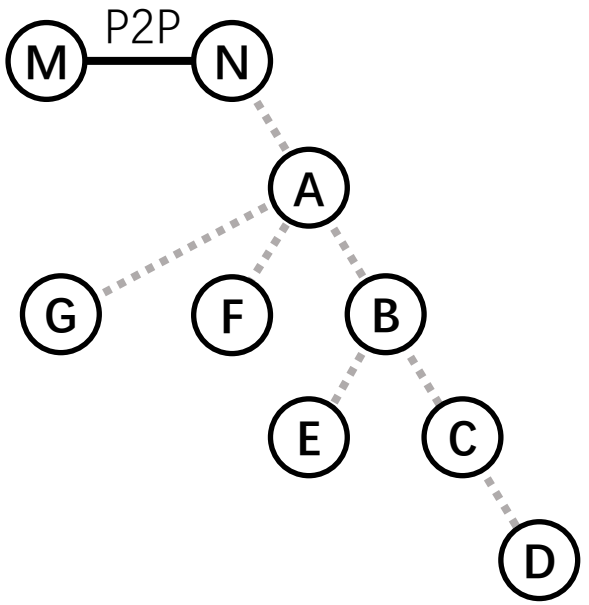
Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order. When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

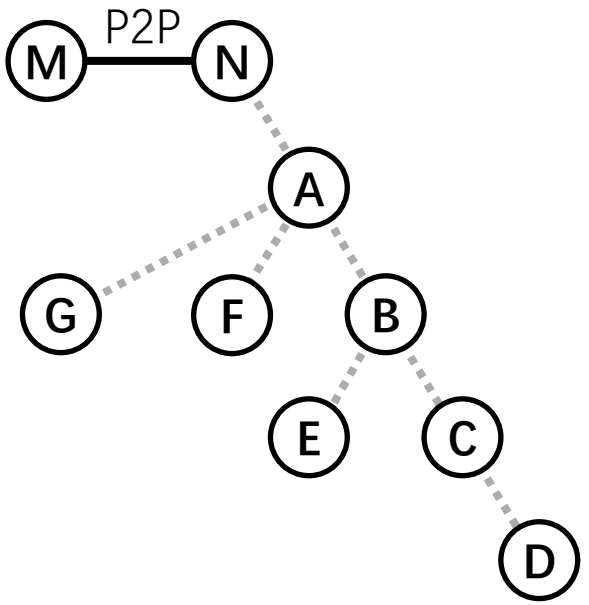
Algorithm 1 AS relationship inference algorithm.

Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order.
When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.
If such a triplet is found, Y-Z is inferred as P2C.



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

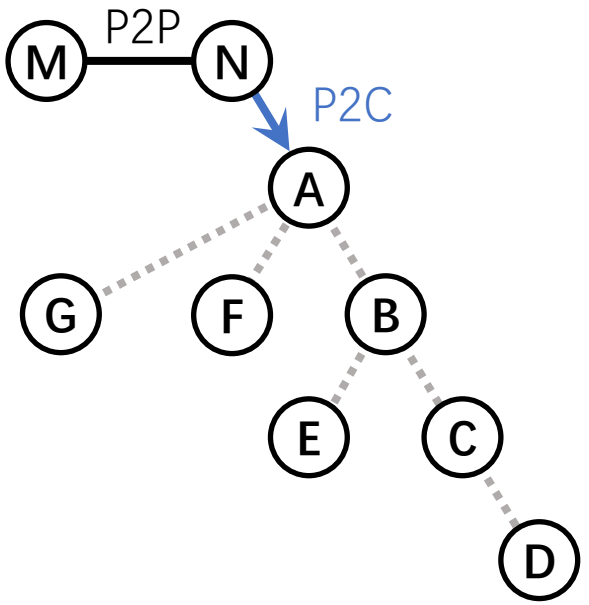
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order.
When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.
If such a triplet is found, Y-Z is inferred as P2C.

- (M N A) implies N-A is P2C



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

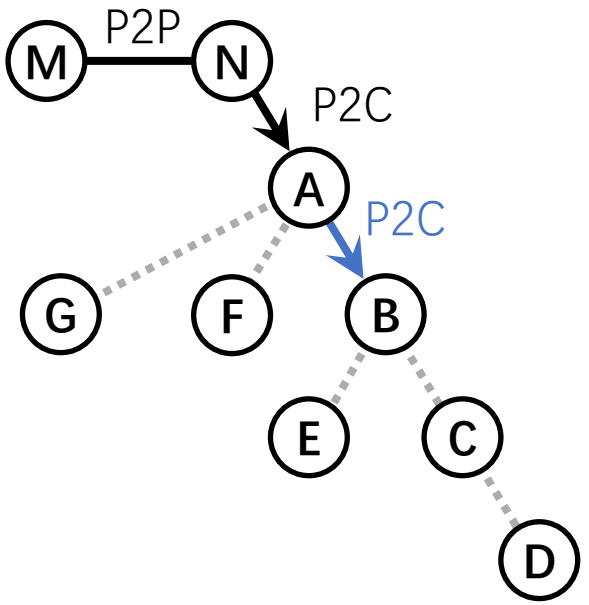
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order.
When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.
If such a triplet is found, Y-Z is inferred as P2C.

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

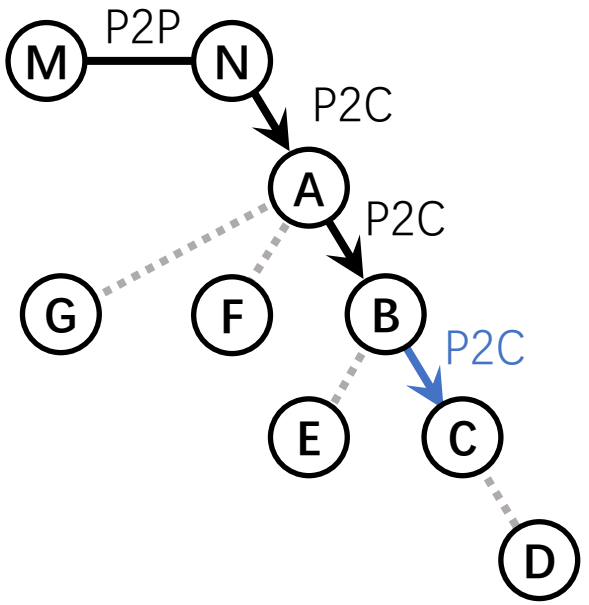
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order.
When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.
If such a triplet is found, Y-Z is inferred as P2C.

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B C) implies B-C is P2C



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

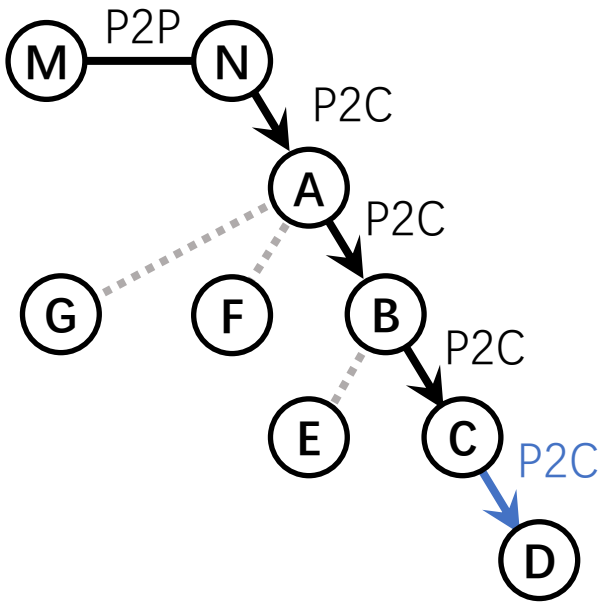
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order.
When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.
If such a triplet is found, Y-Z is inferred as P2C.

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B C) implies B-C is P2C
- (B C D) implies C-D is P2C



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

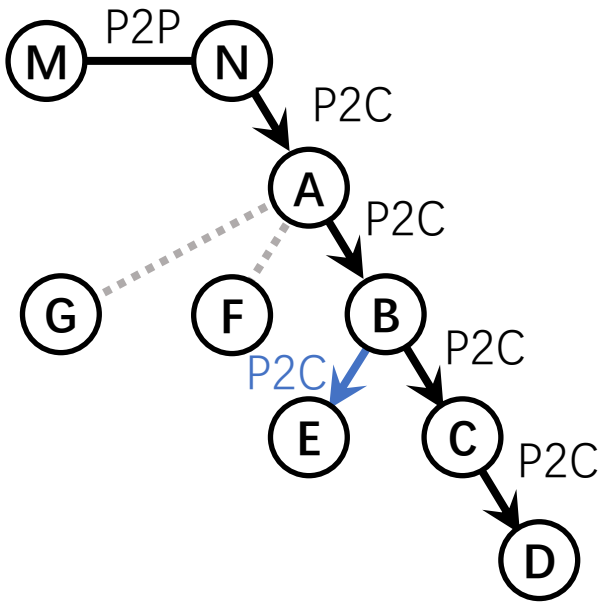
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order.
When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.
If such a triplet is found, Y-Z is inferred as P2C.

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B C) implies B-C is P2C
- (B C D) implies C-D is P2C
- (A B E) implies B-E is P2C



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

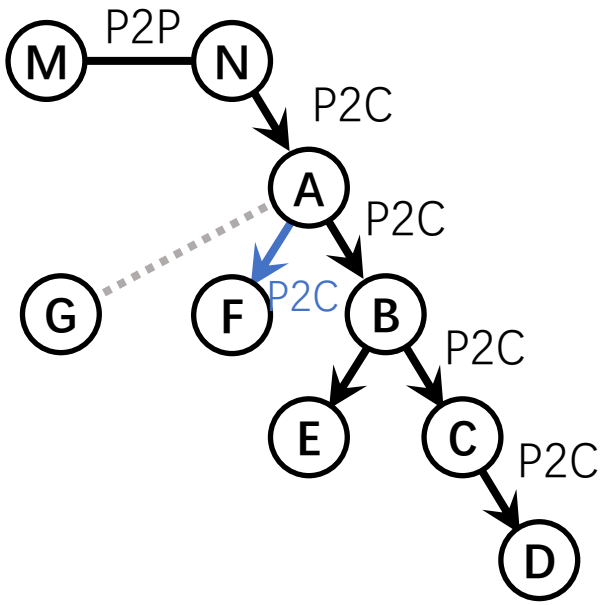
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order.
When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.
If such a triplet is found, Y-Z is inferred as P2C.

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B C) implies B-C is P2C
- (B C D) implies C-D is P2C
- (A B E) implies B-E is P2C
- (N A F) implies A-F is P2C



Resolved links

A Minimal Example (Step 5: infer P2C based on triplets)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

Algorithm 1 AS relationship inference algorithm.

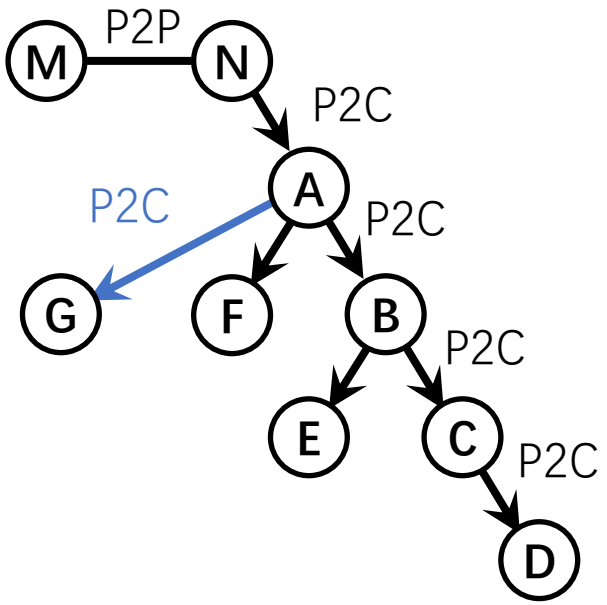
Require: AS paths, Allocated ASNs, IXP ASes
1: Discard or sanitize paths with artifacts (§4.3)
2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
3: Infer clique at top of AS topology (§4.4)
4: Discard poisoned paths (§4.3)
5: Infer c2p rels. top-down using above ranking (§4.5)
6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
8: Infer customers for ASes with no providers (§4.5)
9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)

1. Rank ASes by transit degree
Top-down order: A,B,C,D,E,F,G

2. Infer P2C heuristically

Visit each AS in top-down order.
When visiting Z, find any triplet in the form (X Y Z), where X-Y has already been inferred as P2P or P2C.
If such a triplet is found, Y-Z is inferred as P2C.

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B C) implies B-C is P2C
- (B C D) implies C-D is P2C
- (A B E) implies B-E is P2C
- (N A F) implies A-F is P2C
- (N A G) implies A-G is P2C



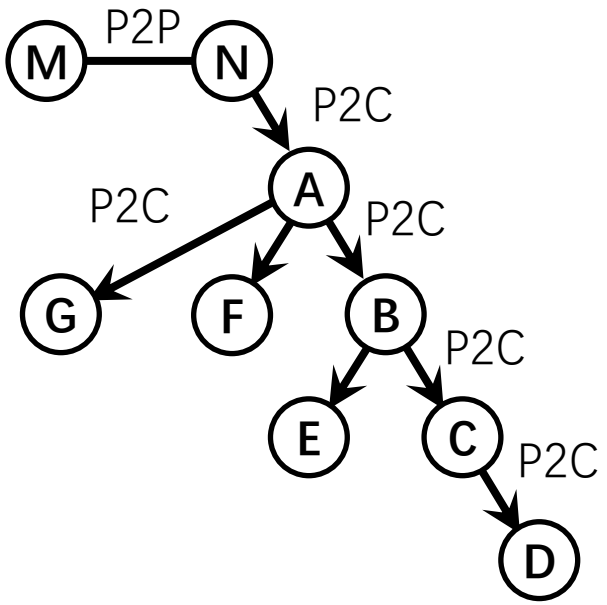
Resolved links

A Minimal Example

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

Empirical input routes

ASRank



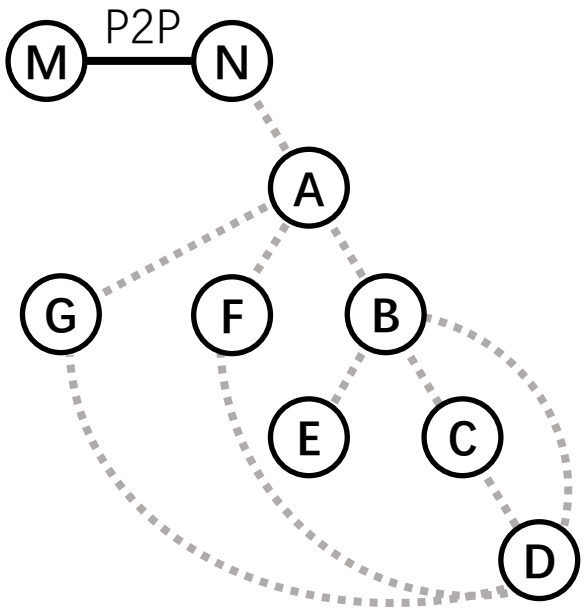
Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M* [*N A B*] *C D*
path 2: ... *M* [*N A B*] *E*
path 3: ... *M* [*N A F*]
path 4: ... *M* [*N A G*]

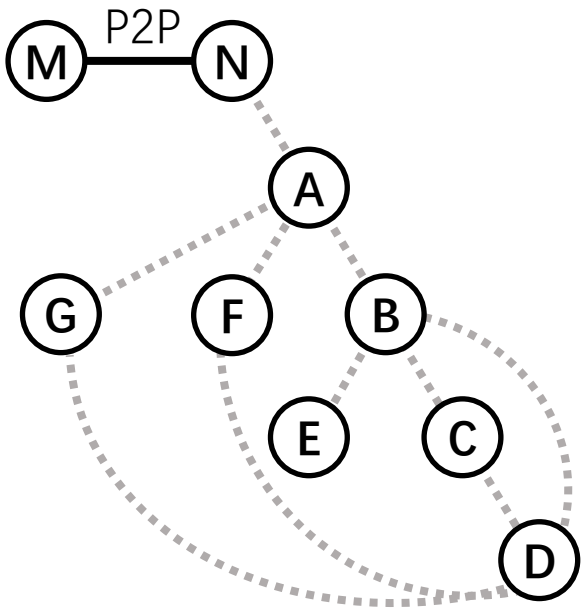
path 5: ... [*X A B*] *D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's individual neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4**+1 = 5**



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

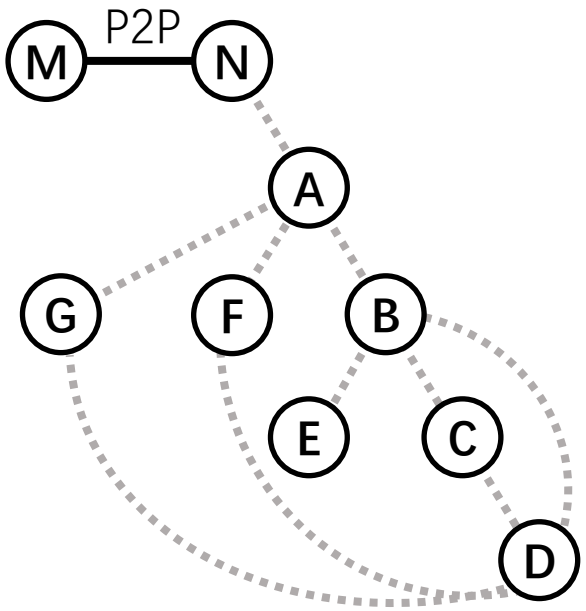
path 5: … *X A B D C Y*
path 6: … *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's individual neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4**+1 = 5**
- TD(B) = 3**+1 = 4**



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

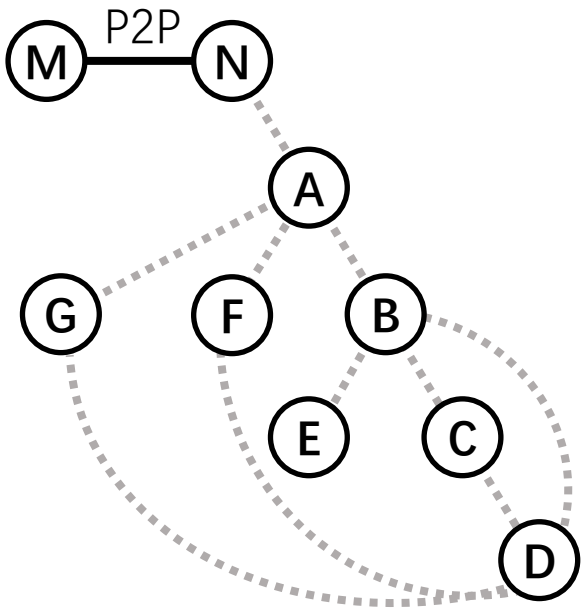
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's individual neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4**+1 = 5**
- TD(B) = 3**+1 = 4**
- TD(C) = 2**+1 = 3**



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

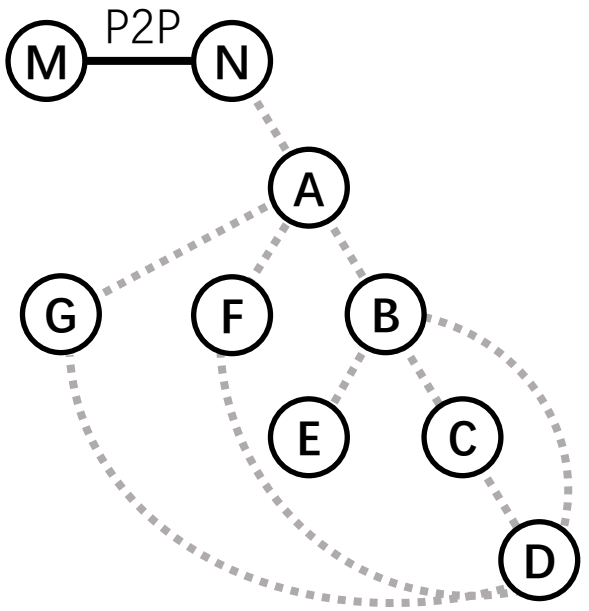
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's individual neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4**+1 = 5**
- TD(B) = 3**+1 = 4**
- TD(C) = 2**+1 = 3**
- TD(D) = 0**+4 = 4**



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

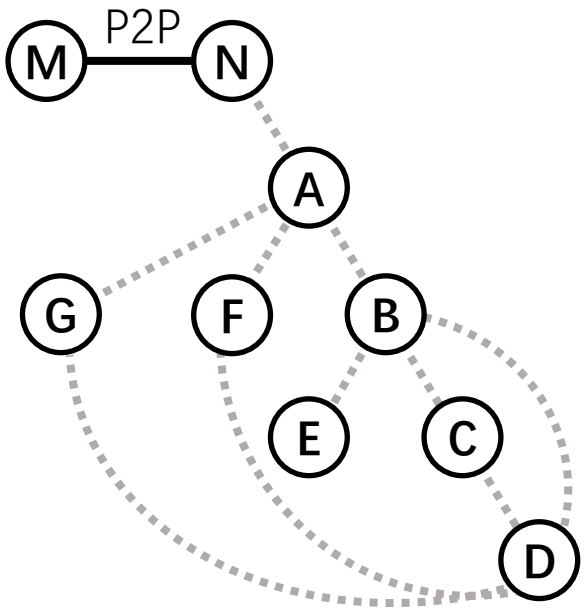
path 5: … *X A B D C Y*
path 6: … *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's individual neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4**+1 = 5**
- TD(B) = 3**+1 = 4**
- TD(C) = 2**+1 = 3**
- TD(D) = 0**+4 = 4**
- TD(E) = 0



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

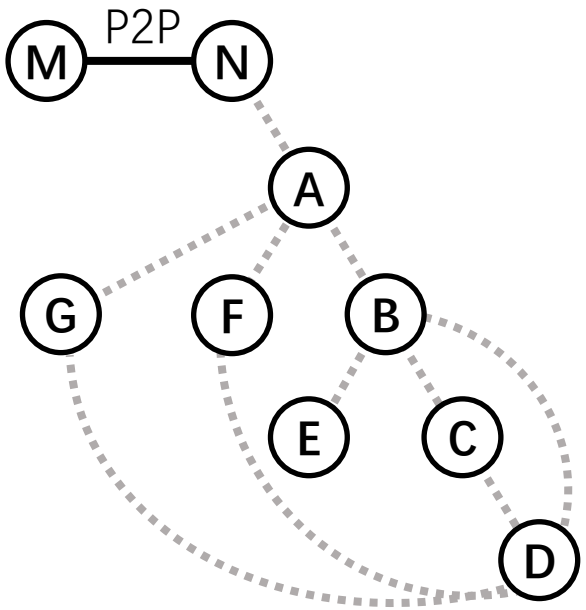
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's individual neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4**+1 = 5**
- TD(B) = 3**+1 = 4**
- TD(C) = 2**+1 = 3**
- TD(D) = 0**+4 = 4**
- TD(E) = 0
- TD(F) = 0**+2 = 2**



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

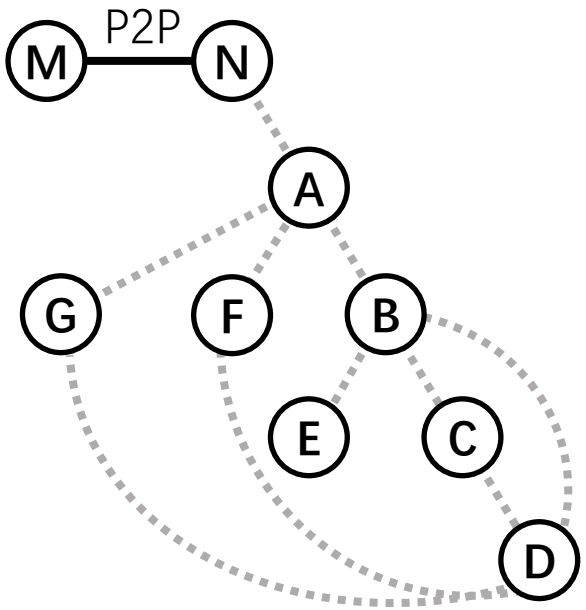
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's individual neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4**+1 = 5**
- TD(B) = 3**+1 = 4**
- TD(C) = 2**+1 = 3**
- TD(D) = 0**+4 = 4**
- TD(E) = 0
- TD(F) = 0**+2 = 2**
- TD(G) = 0**+2 = 2**



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

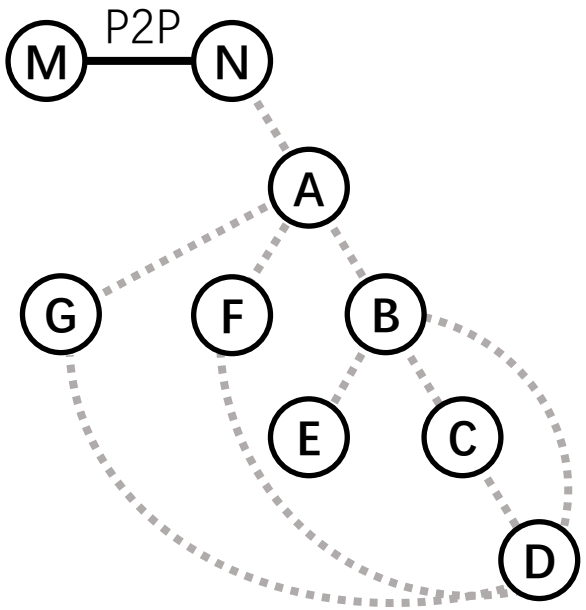
Empirical input routes

1. Rank ASes by transit degree

Transit degree (TD) is the number of subject AS's individual neighbors in AS triplets where the subject AS is in the middle position.

- TD(A) = 4**+1 = 5**
- TD(B) = 3**+1 = 4**
- TD(C) = 2**+1 = 3**
- TD(D) = 0**+4 = 4**
- TD(E) = 0
- TD(F) = 0**+2 = 2**
- TD(G) = 0**+2 = 2**

Top-down order: A,B,**D,C,F,G,E**



Resolved links

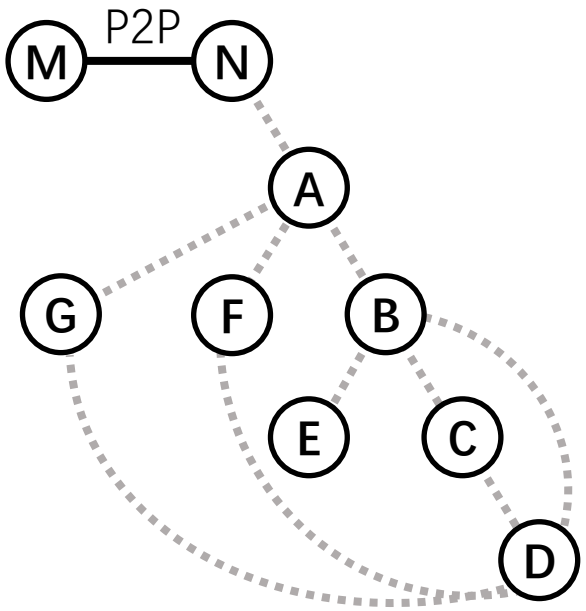
A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**



Resolved links

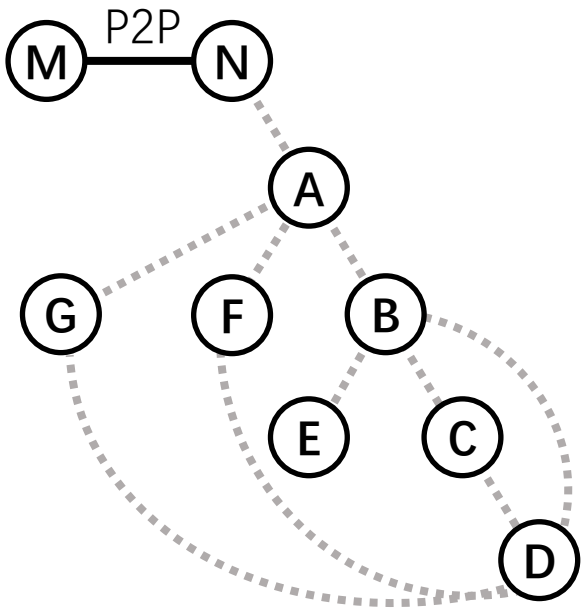
A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. **Rank ASes by transit degree**
   Top-down order: A,B,**D,C,F,G,E**
2. **Infer P2C heuristically**



Resolved links

A Minimal Example (under Adversarial Injection)

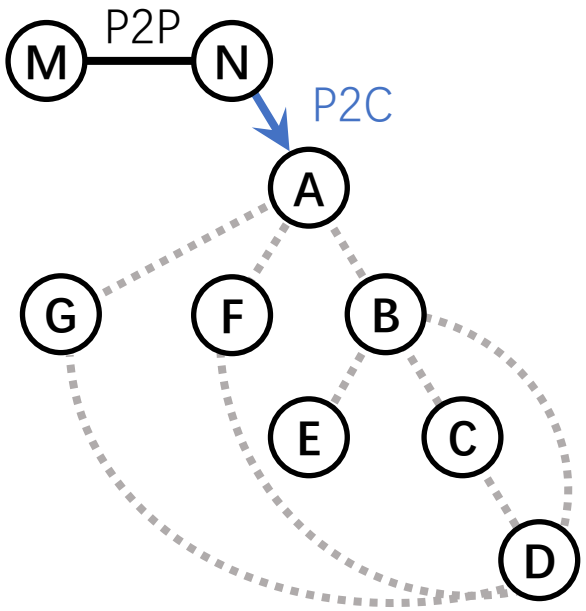
path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**
2. Infer P2C heuristically

- (M N A) implies N-A is P2C



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

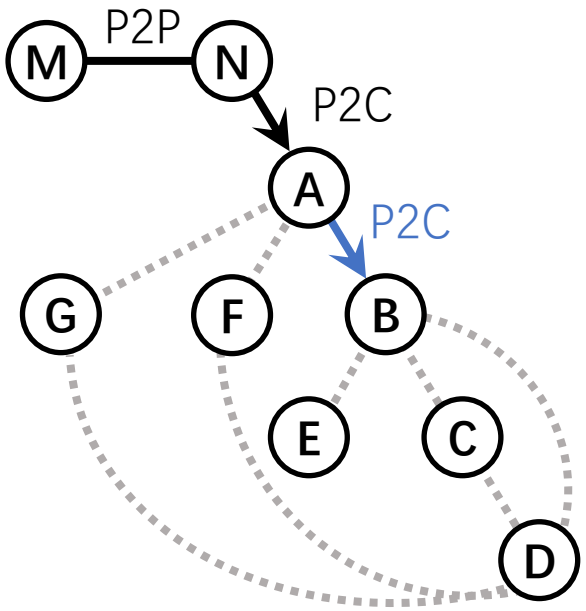
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**

2. Infer P2C heuristically

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

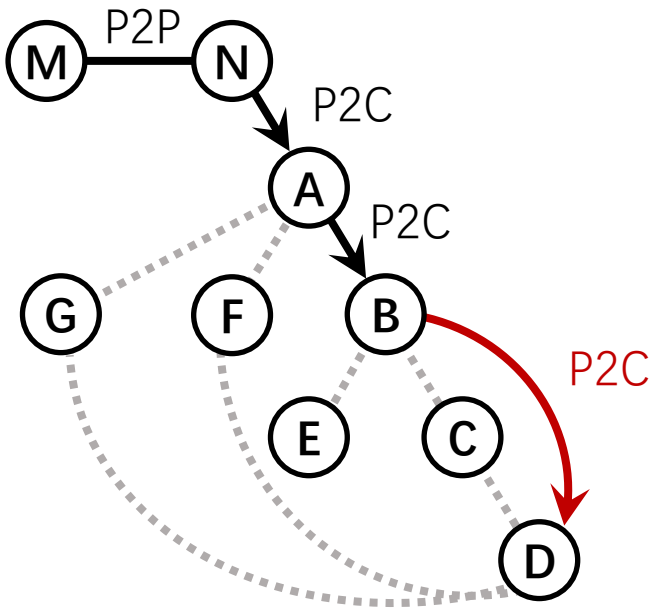
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**

2. Infer P2C heuristically

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B D) implies B-D is P2C



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

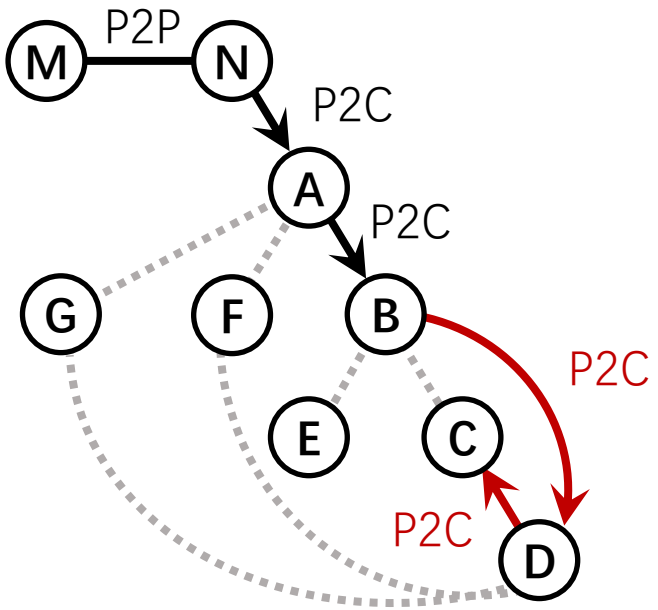
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**

2. Infer P2C heuristically

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B D) implies B-D is P2C
- (B D C) implies D-C is P2C



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

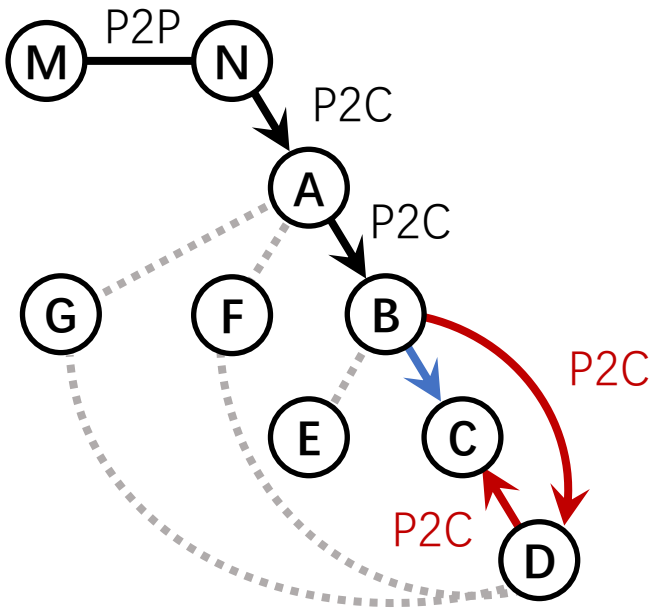
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**

2. Infer P2C heuristically

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B D) implies B-D is P2C
- (B D C) implies D-C is P2C
- (A B C) implies B-C is P2C



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

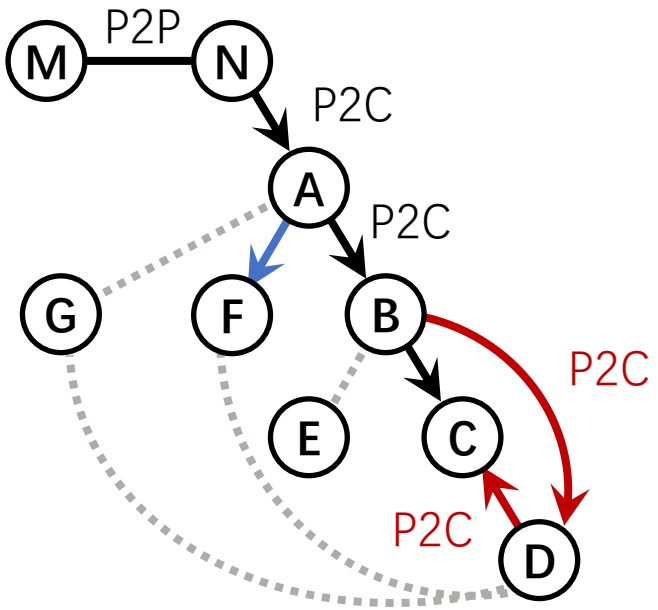
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**

2. Infer P2C heuristically

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B D) implies B-D is P2C
- (B D C) implies D-C is P2C
- (A B C) implies B-C is P2C
- (N A F) implies A-F is P2C



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

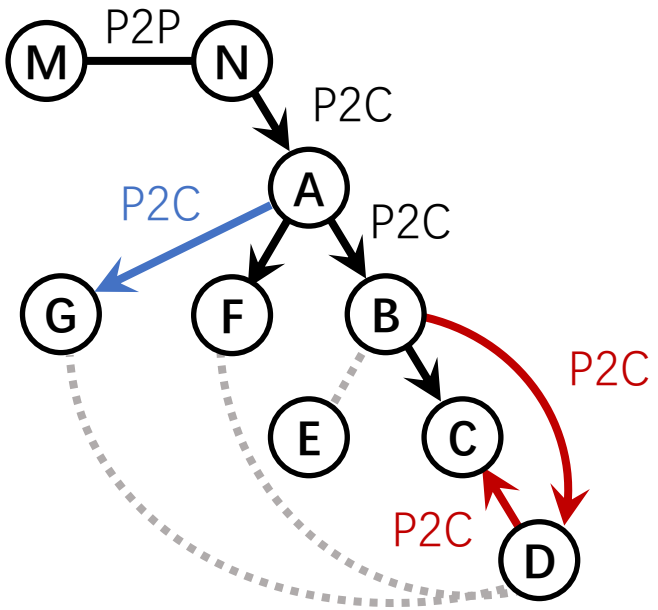
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**

2. Infer P2C heuristically

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B D) implies B-D is P2C
- (B D C) implies D-C is P2C
- (A B C) implies B-C is P2C
- (N A F) implies A-F is P2C
- (N A G) implies A-G is P2C



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

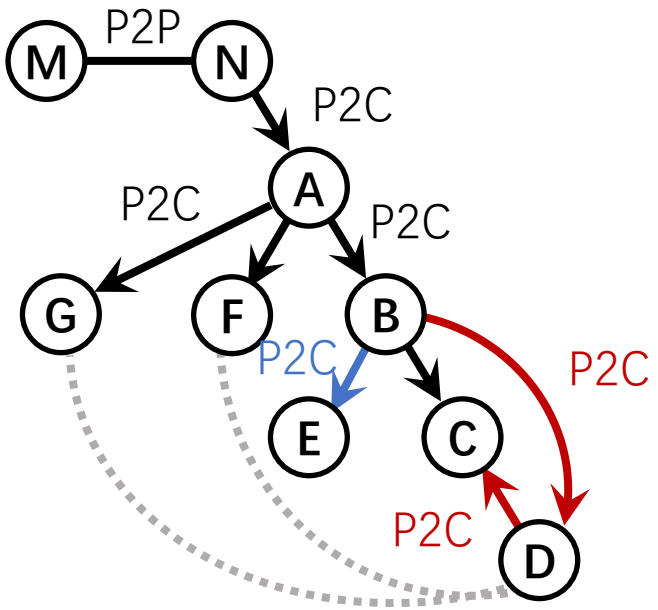
path 5: … *X A B D C Y*
path 6: … *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**

2. Infer P2C heuristically

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B D) implies B-D is P2C
- (B D C) implies D-C is P2C
- (A B C) implies B-C is P2C
- (N A F) implies A-F is P2C
- (N A G) implies A-G is P2C
- (A B E) implies B-E is P2C



Resolved links

A Minimal Example (under Adversarial Injection)

path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

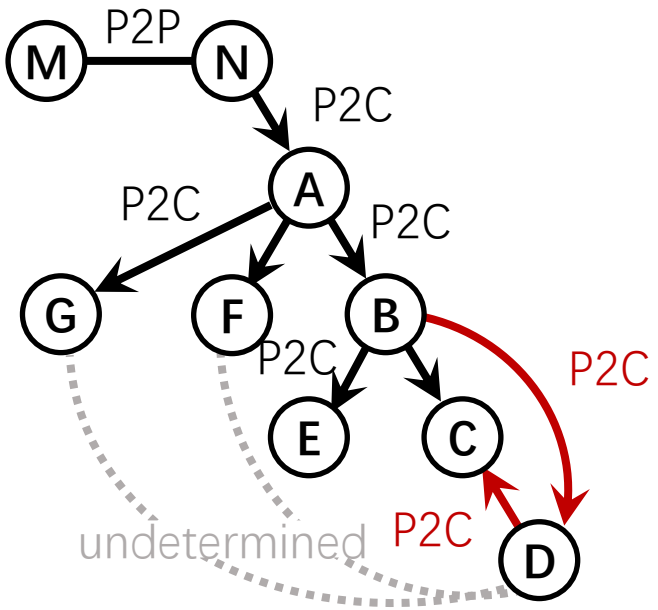
path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes

1. Rank ASes by transit degree
Top-down order: A,B,**D,C,F,G,E**

2. Infer P2C heuristically

- (M N A) implies N-A is P2C
- (N A B) implies A-B is P2C
- (A B D) implies B-D is P2C
- (B D C) implies D-C is P2C
- (A B C) implies B-C is P2C
- (N A F) implies A-F is P2C
- (N A G) implies A-G is P2C
- (A B E) implies B-E is P2C
- G-D and F-D are undetermined as inference information regarding X and Y is not included for simplicity



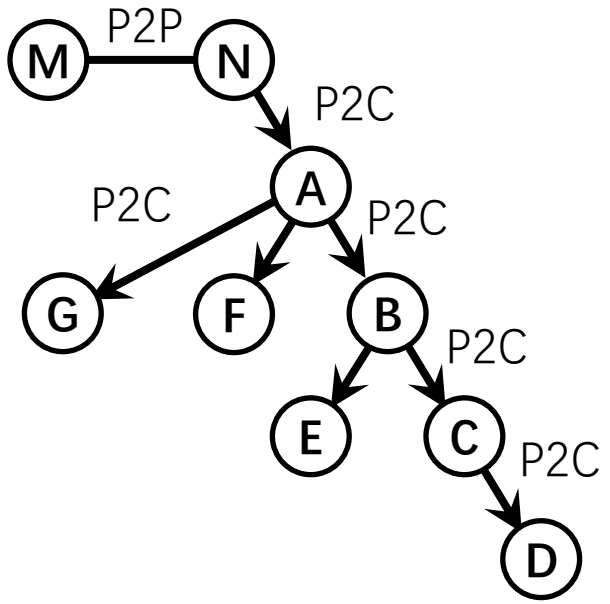
Resolved links

A Minimal Example

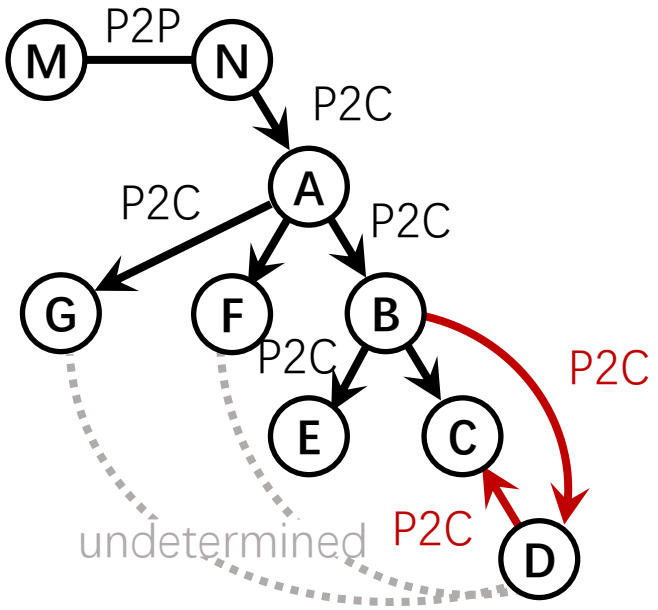
path 1: … *M N A B C D*
path 2: … *M N A B E*
path 3: … *M N A F*
path 4: … *M N A G*

path 5: … *X A B D C Y*
path 6: … *X F D G Y*

Empirical input routes



Resolved links (baseline)



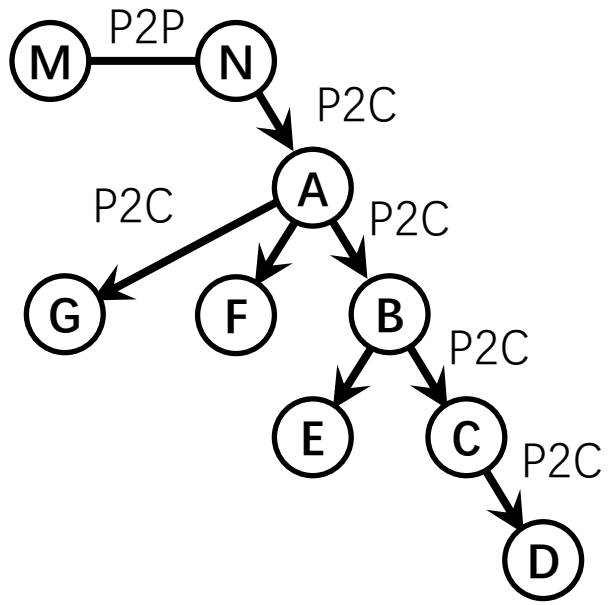
Resolved links (manipulated)

A Minimal Example

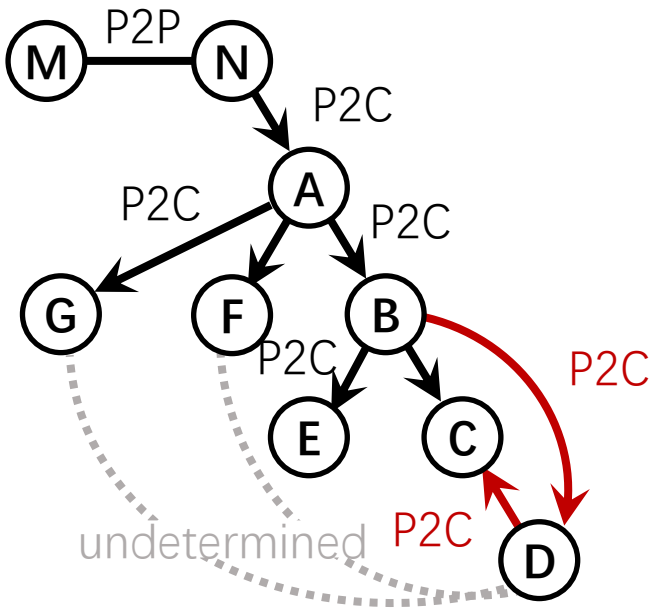
path 1: ... *M N A B C D*
path 2: ... *M N A B E*
path 3: ... *M N A F*
path 4: ... *M N A G*

path 5: ... *X A B D C Y*
path 6: ... *X F D G Y*

Empirical input routes



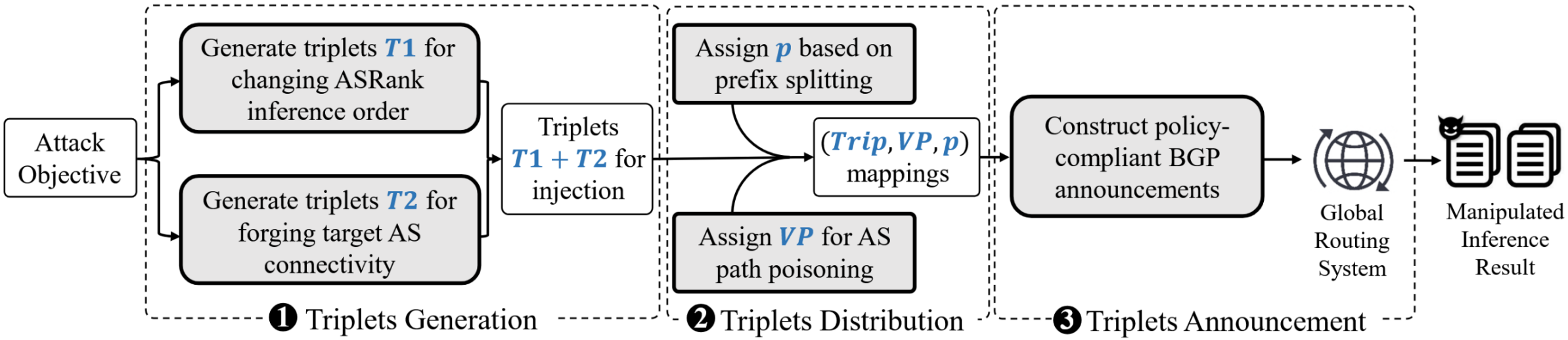
Resolved links (baseline)



Resolved links (manipulated)

One can craft and inject such forged paths via policy-compliant BGP announcements to induce targeted manipulation in a principled way!

A Manipulation Attack Against ASRank



Offline evaluation on a mirrored ASRank pipeline:

- 100 crafted triplets can manipulate nearly 50% of all inferred relationships.
- Only 10-30 announcements can flip over 80–90% of P2C relationships involving small providers.

Controlled experiment on PEERING testbed:

- AS triplets embedded in forged routes can be propagated and observed globally.

Discussions

Mitigations

- Randomize ASRank inference schedule
- Ensemble or multi-run consensus inference
- Longitudinal consistency monitoring
- Cross-dataset sanity checking
- Link/path validation mechanisms (ASPA, Peerlock, etc.)

Broader insights

- Trustworthiness of derived topology knowledge
- Measurement manipulation as a systemic risk
- Security-measurement feedback loop

Q&A

Thank you!

Contact: yh-chen21@mails.tsinghua.edu.cn (Yihao Chen)

Feel free to request a copy of the slides and our full paper!