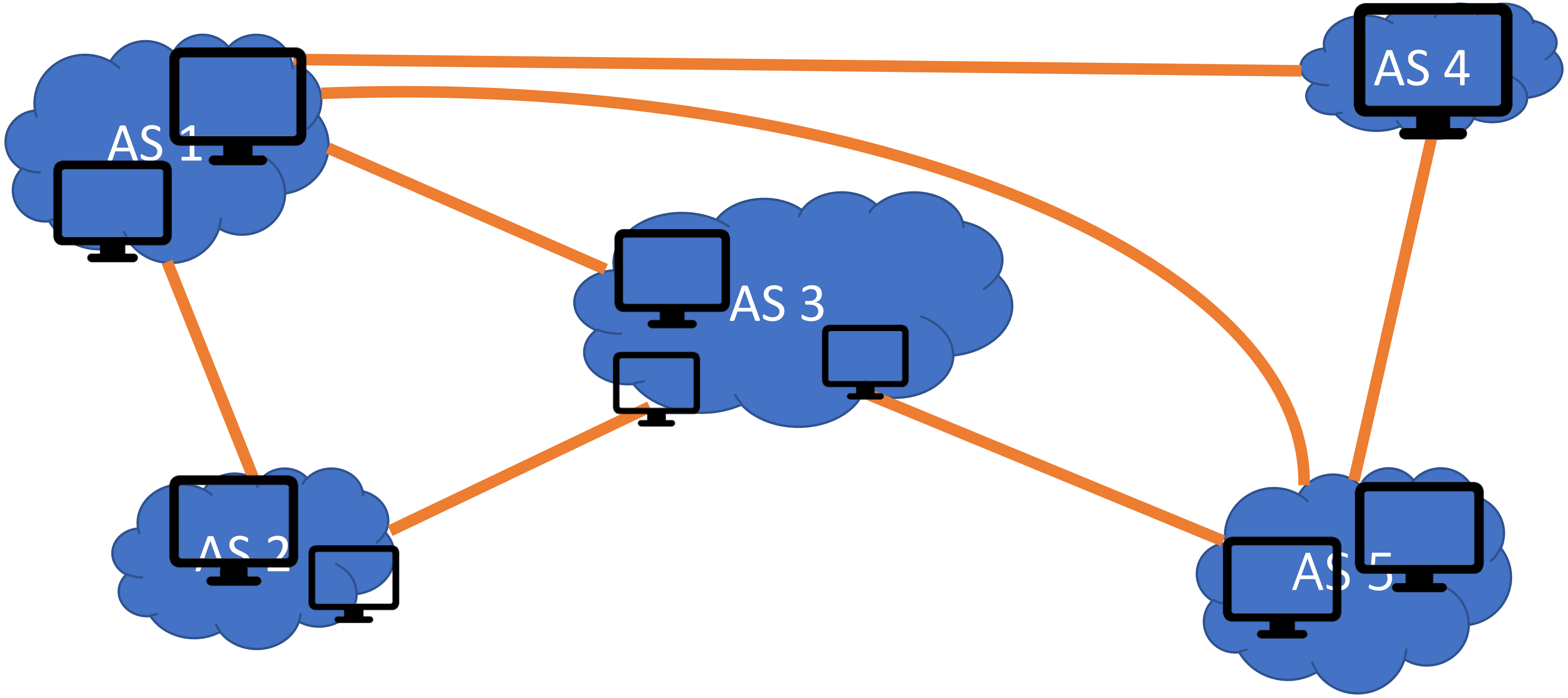
# Stable and Practical AS Relationship Inference with ProbLink

**Yuchen Jin**[1] , Colin Scott[2], Amogh Dhamdhere[3],
Vasileios Giotsas[4], Arvind Krishnamurthy[1], Scott Shenker[2]

[1]University of Washington, [2]UC Berkeley, [3]CAIDA, [4]Lancaster University

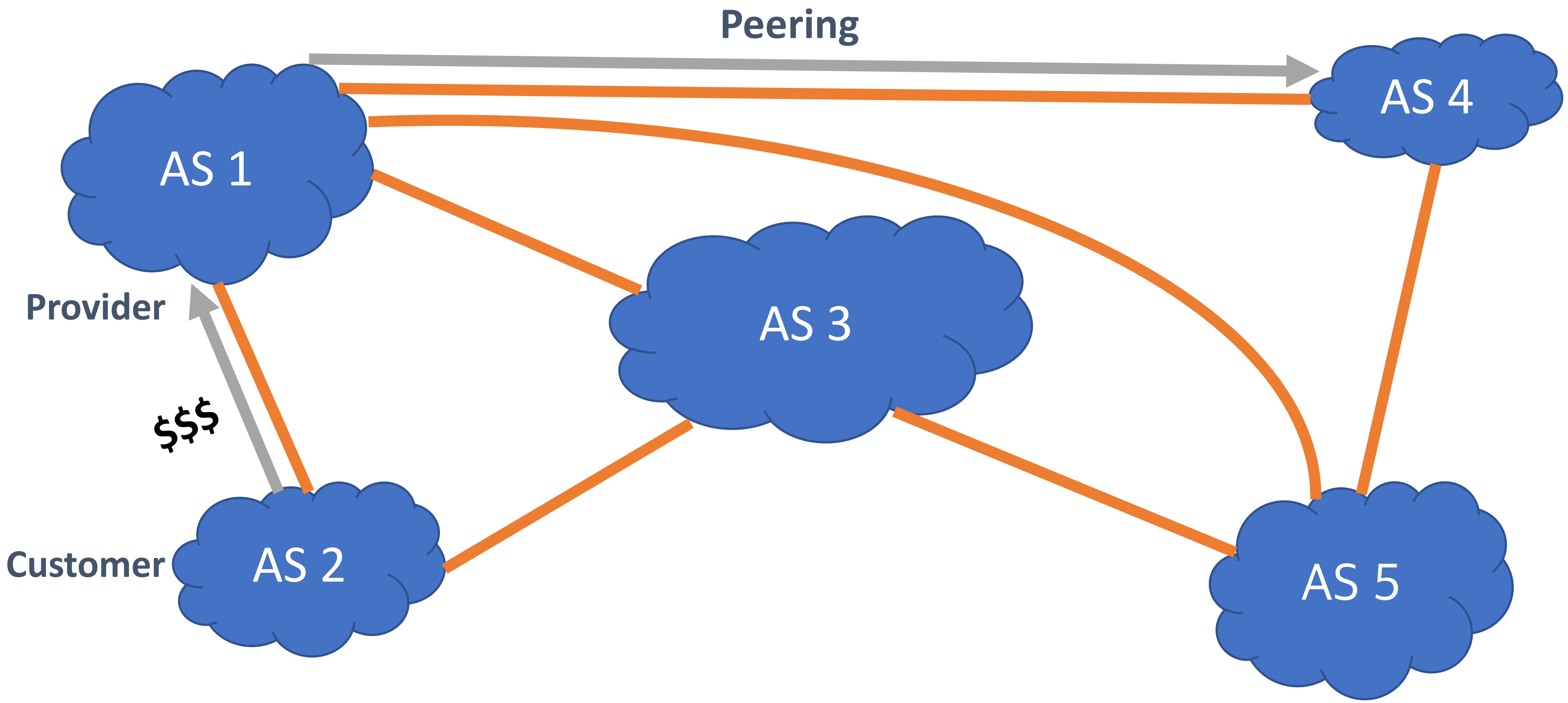
# Autonomous System (AS) and BGP

**ASes:**
Google
Comcast
UW



# AS Relationships

**ASes keep relationships confidential!**




| | |
|---|---|
| **Customer-to-provider (Cust-Prov)** | Customer AS pays the provider AS for transit |
| **Peer-to-peer (Peer-Peer)** | Settlement free |

# Motivation

- AS relationship inference application domains
  - Understanding Internet evolution
  - Identifying malicious ASes
  - Detecting network congestion

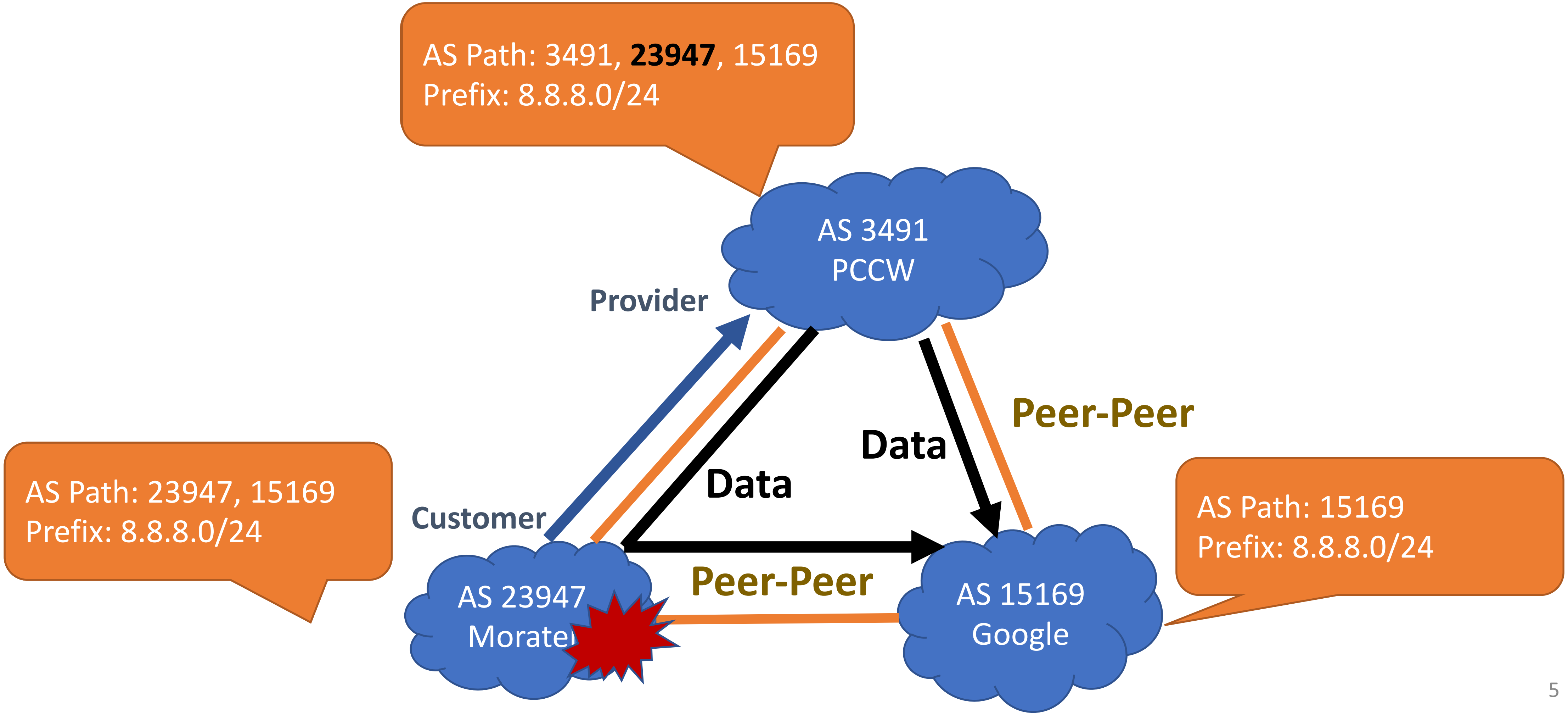
- A research problem which has been studied for ~2 decades
  - Gao (2001)
  - Subramanian et al. (2002)
  - Di Battista et al. (2003)
  - Dimitropoulos et al. (2007)
  - ***AS-Rank*** (2013)

**99% accuracy (1% error rate)**

How does *AS-Rank* perform for actual applications?

# Case Study: Route Leak Detection

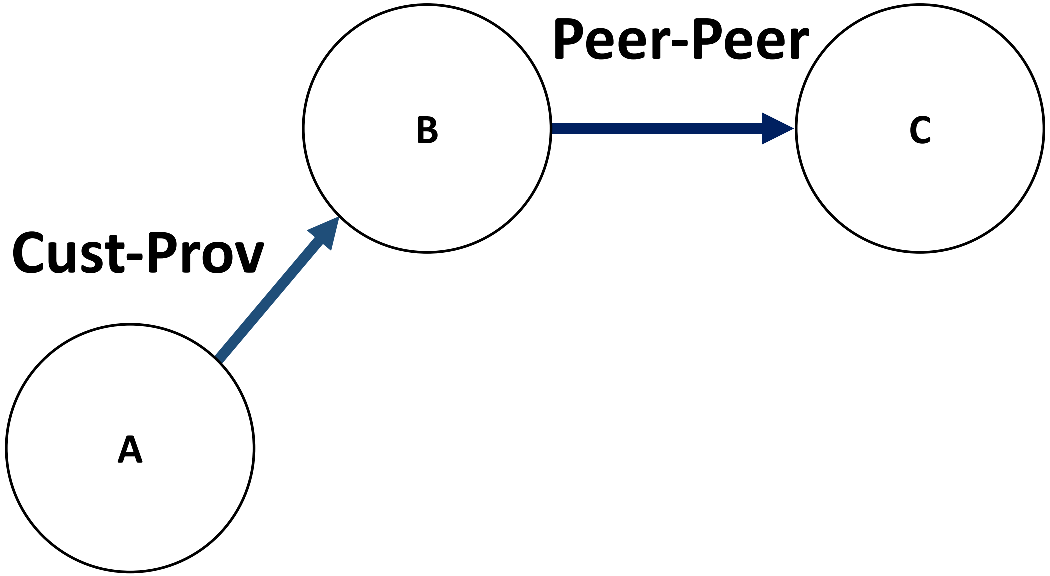
**On November 5, 2012, Google's services cannot be accessible in Asia for half an hour.**



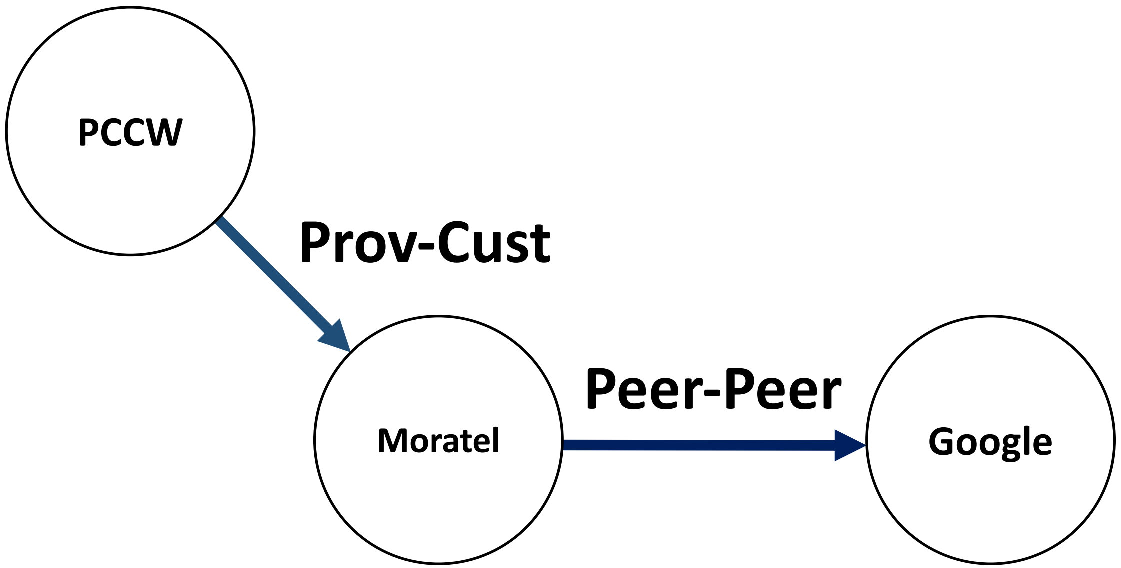
# Case Study: Route Leak Detection

**Valley-free assumption:**
Path consists of Cust-Prov links, followed by zero or one Peer-Peer link, followed by Prov-Cust links. (uphill – cross – downhill)

**Valley-free:**




**Valley-free violation:**



# Route Leak Detection

## Detection method: Check valley-free violations

## AS relationship validation dataset:

- relationships encoded using the Community attribute in BGP paths
  - E.g., AS209 (CenturyLink) uses the Community 13570 to tag routes received from customers

# Route Leak Detection Using AS-Rank

Detect route leaks using *AS-Rank*:

- Low precision: only **20%** of the route leaks detected using *AS-Rank* were real route leaks
- Low recall: **78%** of the real leaks were missed

# Outline

- *AS-Rank* does not meet the demands of actual applications.

- We develop a simple inference algorithm *CoreToLeaf* that achieves accuracy comparable to *AS-Rank*.
  - Most links in the validation dataset are relatively easy to infer.

- We identify different subsets of the validation dataset that are **hard** to infer.

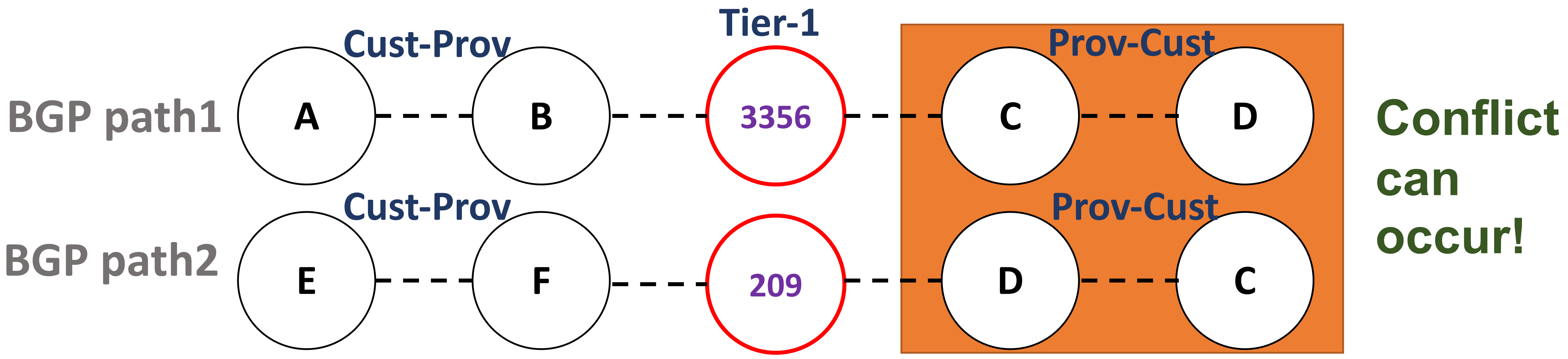
- We develop **ProbLink** - a probabilistic AS relationship inference algorithm.

- Evaluation

# A Simple Algorithm: CoreToLeaf

1. Infer a transit-free clique (i.e., Tier-1) ASes.

2. For each path that traverses a Tier-1:




3. Label all remaining unclassified links as Peer-Peer.

# CoreToLeaf vs. AS-Rank

**April 1st, 2017**

| Algorithm | Validation dataset Precision (%) | Validation dataset Recall (%) | Conflict (%) | Route Leak Detection Precision (%) | Route Leak Detection Recall (%) |
|---|---|---|---|---|---|
| CoreToLeaf | 97.0 | 97.3 | 0.12 | 19.8 | 22.1 |
| AS-Rank | 98.4 | 98.6 | 0 | 8.1 | 5.6 |

**Implications:**

1. Most links are easy to infer
2. Need to focus on ***hard*** links

# Extract *Hard* Links

- Use gradient boosting decision tree to calculate feature importance

| Category | *CoreToLeaf* error (%) | *AS-Rank* error (%) |
|---|---|---|
| **Max node degree < 100** | 13.7 | 8.6 |
| **Observed by 50-100 VPs** | 4.7 | 9.3 |
| **Non-VP & Non-Tier1** | 5.3 | 9.0 |
| **Unlabeled Stub-clique** | 95.5 | 33.4 |
| **Conflict** | 100 | 8.1 |

**April 1st, 2017**

- Fraction of *hard* links in the **validation dataset** are **3X** fewer than that in the **overall links**

→ **The validation dataset is skewed to easy links**

# Outline

- *AS-Rank* does not meet the demands of actual applications.
- We develop a simple inference algorithm *CoreToLeaf* that achieves accuracy comparable to *AS-Rank*.
  - Most links in the validation dataset are relatively easy to infer.
- We identify different subsets of the validation dataset that are hard to infer.
- We develop **ProbLink** - a probabilistic AS relationship inference algorithm.
- Evaluation

# Feature Design

(A) The structure of paths that use the link

- Triplet feature

(B) The structure of paths that **do not** use the link

- Non-path feature

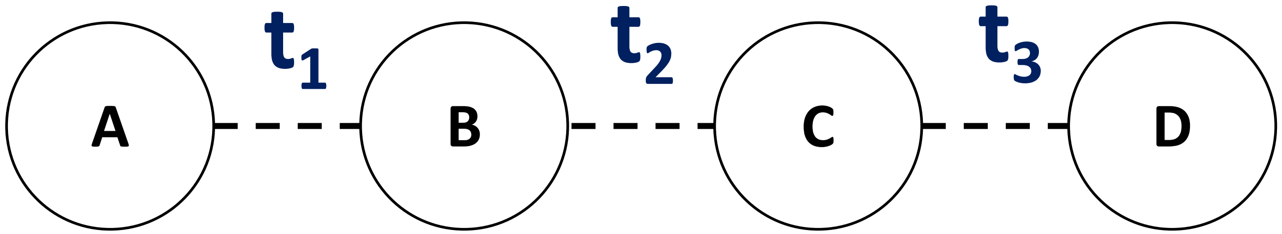
(C) Connectivity properties of the link

- Distance to clique
- Number of VPs observing a link
- Co-located IXP/peering facility

# Triplet Feature

**Intuition:** Most triplets are valley-free compliant, but we should tolerate some valleys.

➢ Likelihood of valleys is derived from data




**$t \in$ {Cust-Prov, Prov-Cust, Peer-Peer}**

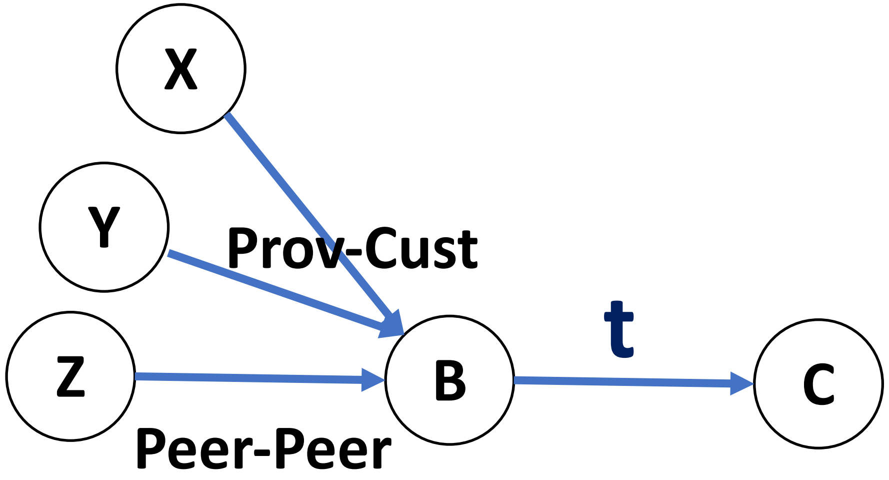
**Definition:** Attribute probabilistic values for the relationships of the first and the last links given the relationship of the middle link – $P(t_1, t_3 | t_2)$

# Non-path Feature

**Intuition**: If none of the Peer-Peer/Prov-Cust links coming into an AS are followed by a specific link, then the link is likely not a Prov-Cust link.

**Definition:** Attribute probabilistic values for the number of previous Peer-Peer/Prov-Cust links given the relationship of the next link.

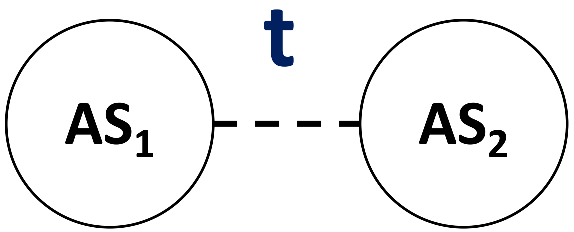
– **P(# Peer-Peer + # Prov-Cust | t)**




**No path contains:**

- X-B-C
- Y-B-C
- Z-B-C

# Connectivity Features




Distance to clique feature: **P(dist($AS_1$), dist($AS_2$) | t)**

**Intuition:**

- High-tier ASes are closer to clique ASes than low-tier ASes
- Peer-Peer links are typically between the ASes in the same tier

Vantage point feature: **P(# VPs observing $AS_1$ -- $AS_2$| t)**

**Intuition:** Prov-Cust links are more likely to be seen by more VPs

Co-located IXP and co-located peering facility feature:

(extracted from PeeringDB) **P(# IXPs/facilities| t)**

**Intuition:** The more IXPs or facilities two ASes are co-located in, the more likely they are peering with each other

# ProbLink Algorithm

Use *CoreToLeaf* to come up with an initial labeling

***Repeat***:

- Compute the conditional probability distribution of each feature based on current labels
  - $P(f_i \mid t)$ E.g., P(# VPs observing a link| t)
- Predict new label for each link using Naïve Bayes and conditional probability distributions
  - $$\hat{t} = \arg\max_t P(t) \prod_{i=1}^{n} P(f_i \mid t)$$

***Stop upon convergence***

# Outline

- *AS-Rank* does not meet the demands of actual applications.

- We develop a simple inference algorithm *CoreToLeaf* that achieves accuracy comparable to *AS-Rank*.
  - Most links in the validation dataset are relatively easy to infer.

- We identify different subsets of the validation dataset that are hard to infer.

- We develop **ProbLink** - a probabilistic AS relationship inference algorithm.

- Evaluation

# Accuracy Comparison

How well can *ProbLink* perform across years?

Compare inference error rates to that of *AS-Rank:*

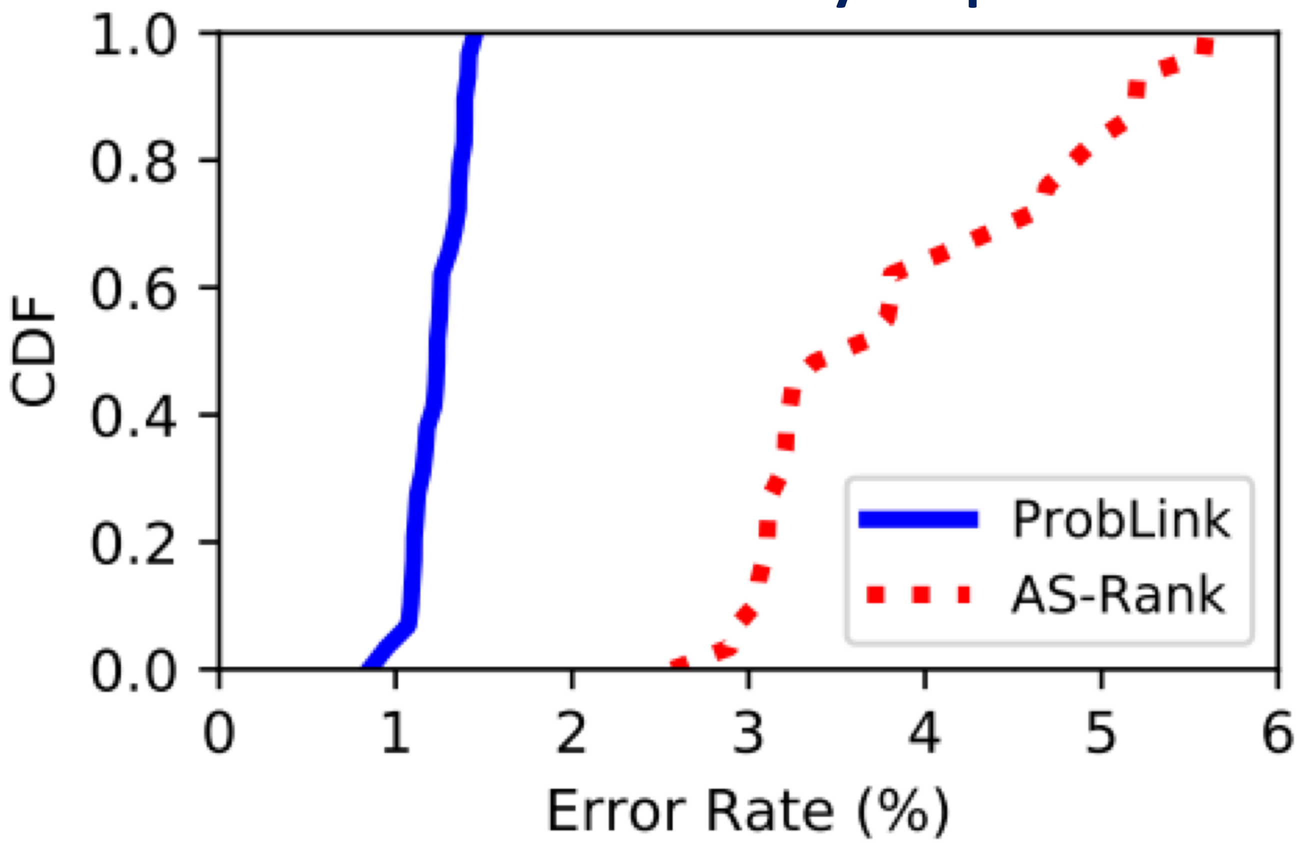
- The whole validation dataset: **1.7X** improvement
- ***Hard*** links: **1.8X** to **6.1X** improvement

| Category | *AS-Rank* error (%) | *ProbLink* error (%) | Improvement |
|---|---|---|---|
| **Observed by 50-100 VPs** | 8.8 | 1.5 | 5.9X |
| **Non-VP & Non-Tier1** | 4.4 | 1.7 | 2.6X |
| **Unlabeled Stub-clique** | 33.6 | 5.5 | 6.1X |
| **Conflict** | 6.8 | 3.8 | 1.8X |
| **Max node degree < 100** | 8.6 | 4.4 | 2.0X |

# Stability Analysis

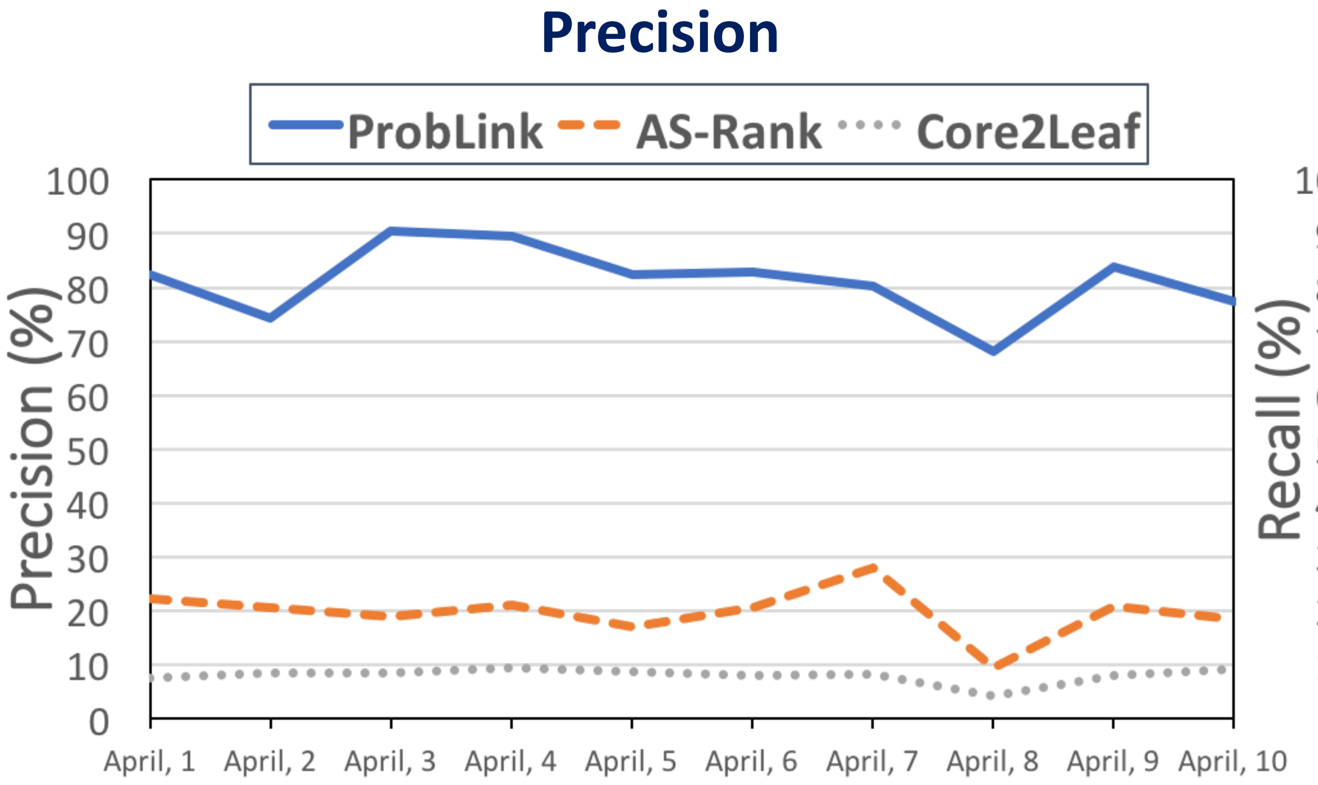
- *AS-Rank* is sensitive to snapshot selection
- *ProbLink* is consistently stable

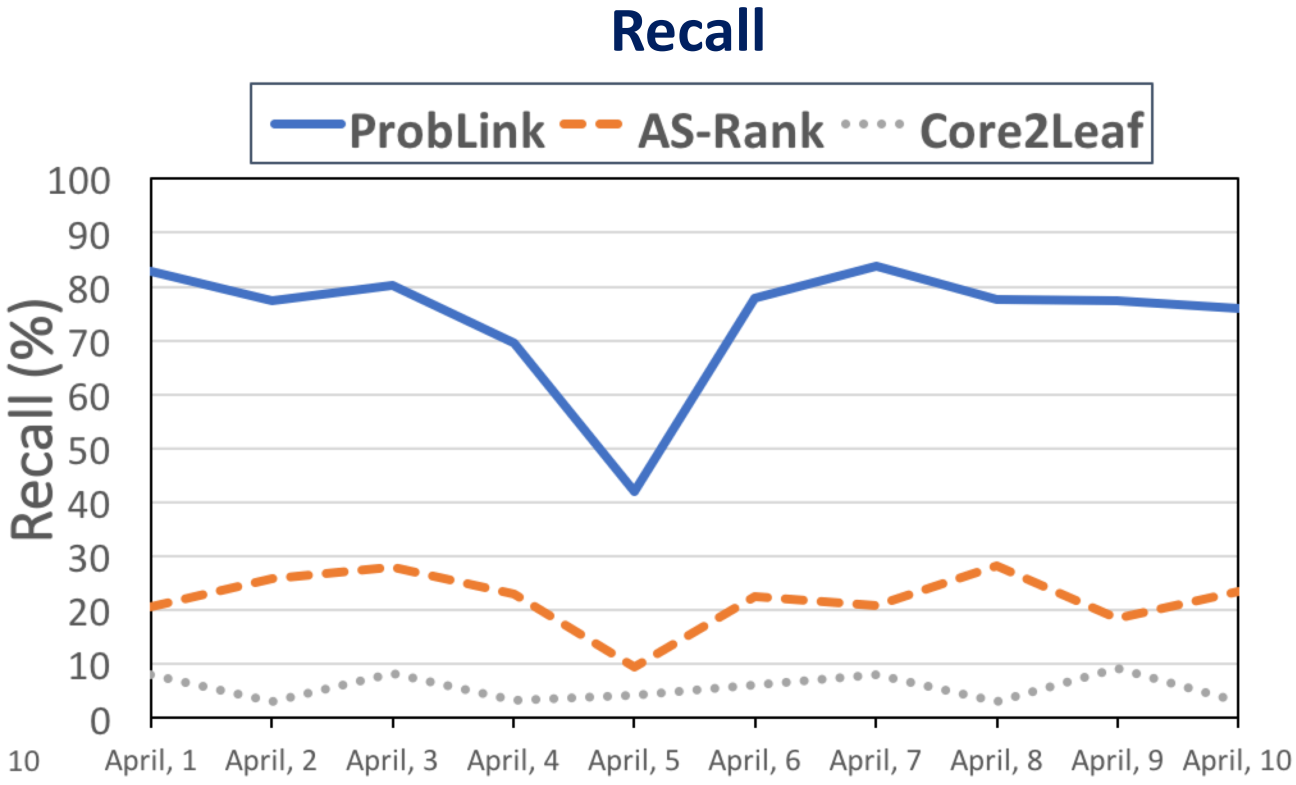
**30 consecutive 1-day snapshots**



# Route Leak Detection

**Route leak detection method: Check valley-free violations**




| ProbLink | 81.1% |
|---|---|
| AS-Rank | 19.8% |
| CoreToLeaf | 8.1% |

| ProbLink | 76.2% |
|---|---|
| AS-Rank | 22.1% |
| CoreToLeaf | 5.6% |

# Practical Applications

Improvements over AS-Rank:

- Route leak detection
  - Precision: 4.1X
  - Recall: 3.4X
- Complex relationship inference
  - Reveals 27% more complex relationships
- Predicting the impact of selective advertisement
  - Increases the precision by 34%

# Conclusion

- High accuracy in validation dataset does not translate to high application-level performance
- Most links in the validation dataset are easy to infer
- Constructed hard links sets and use them as benchmarks
- Developed ProbLink and allow for integration of many noisy but useful attributes
- Demonstrated that ProbLink is more effective and stable for real applications than previous techniques