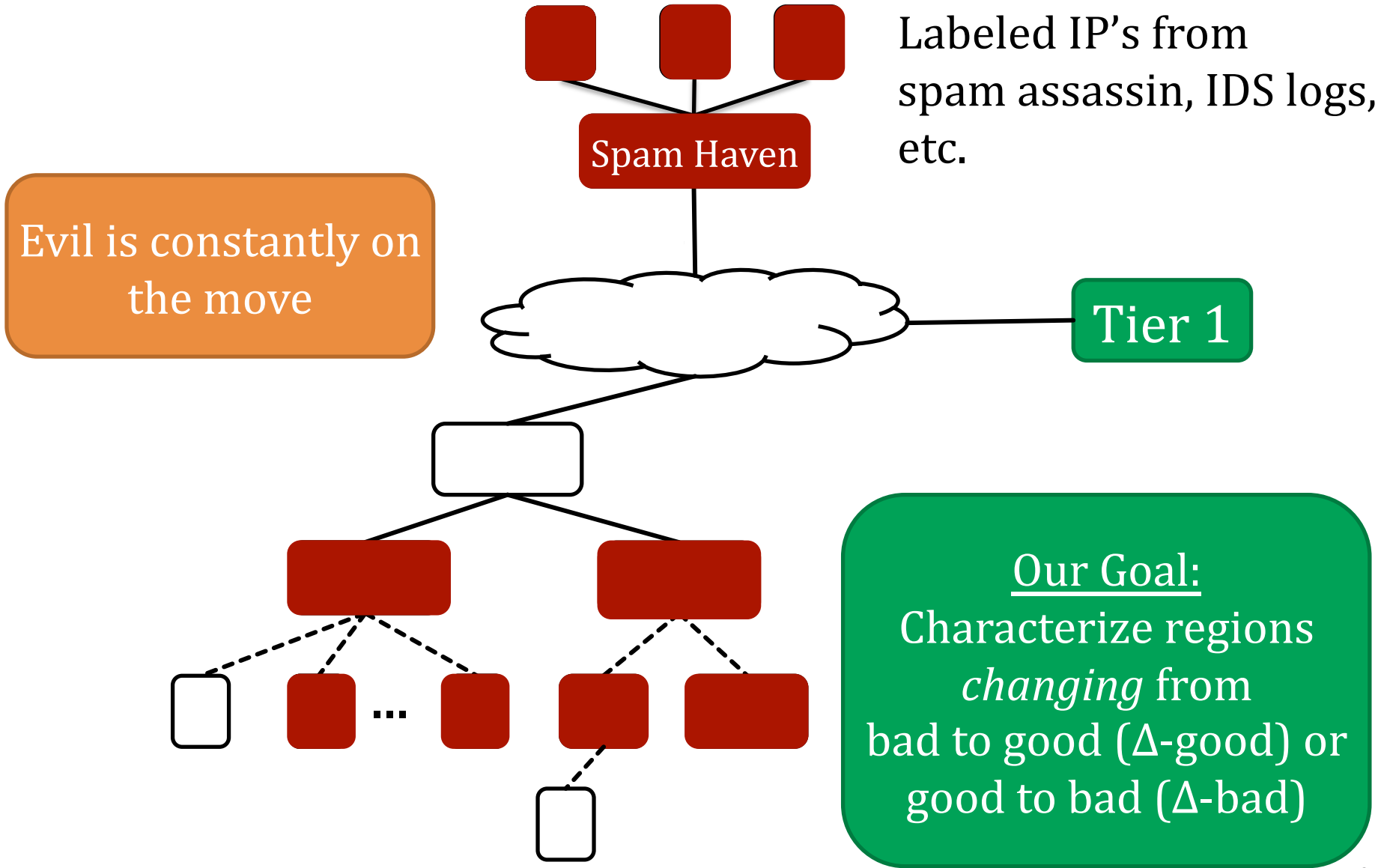
Automatically Inferring the Evolution of Malicious Activity on the Internet

Shobha Venkataraman
AT&T Research

David Brumley
Carnegie Mellon University

Subhabrata Sen
AT&T Research

Oliver Spatscheck
AT&T Research



Labeled IP's from spam assassin, IDS logs, etc.

Evil is constantly on the move

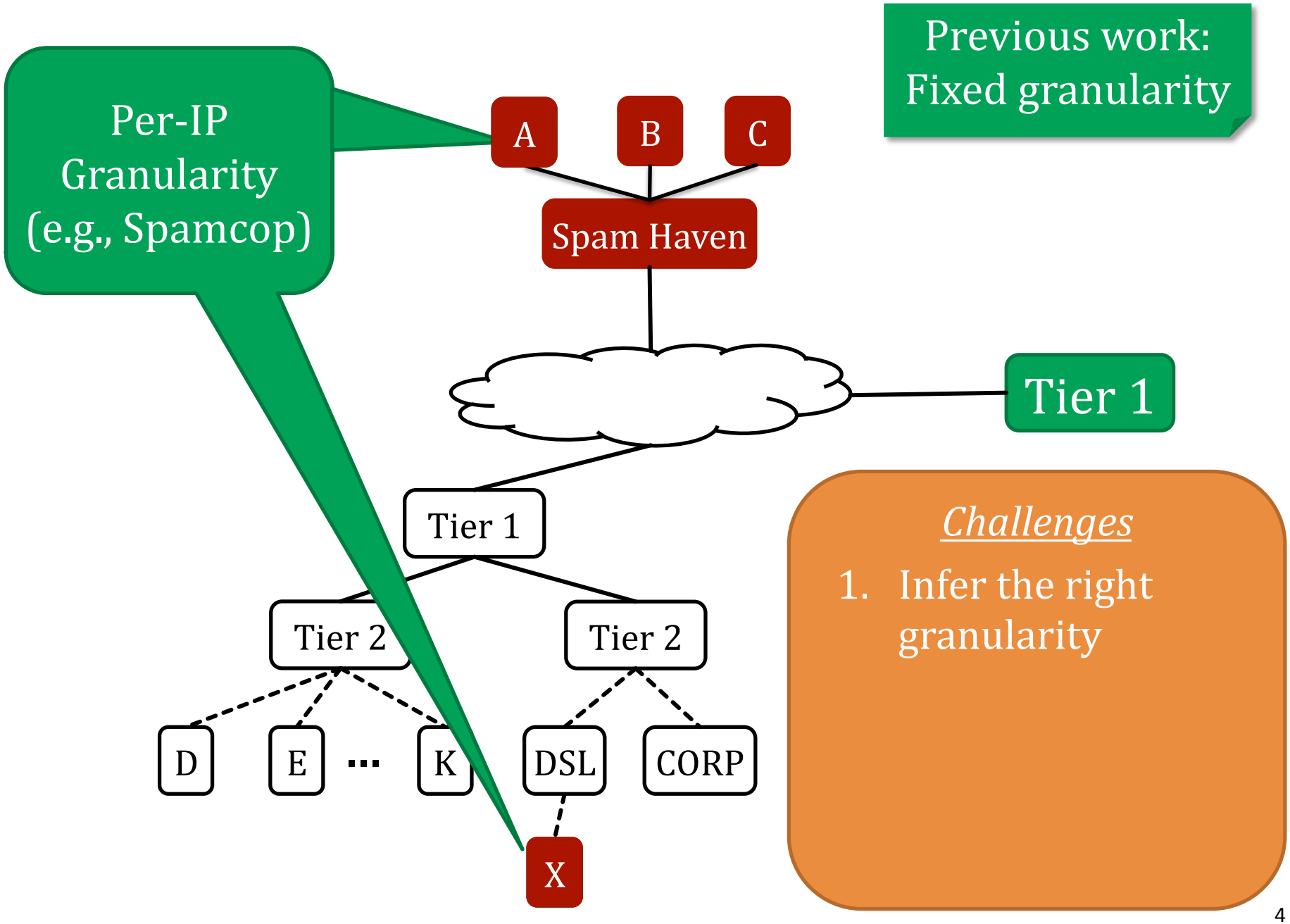
Our Goal:
Characterize regions *changing* from bad to good (Δ-good) or good to bad (Δ-bad)

Research Questions

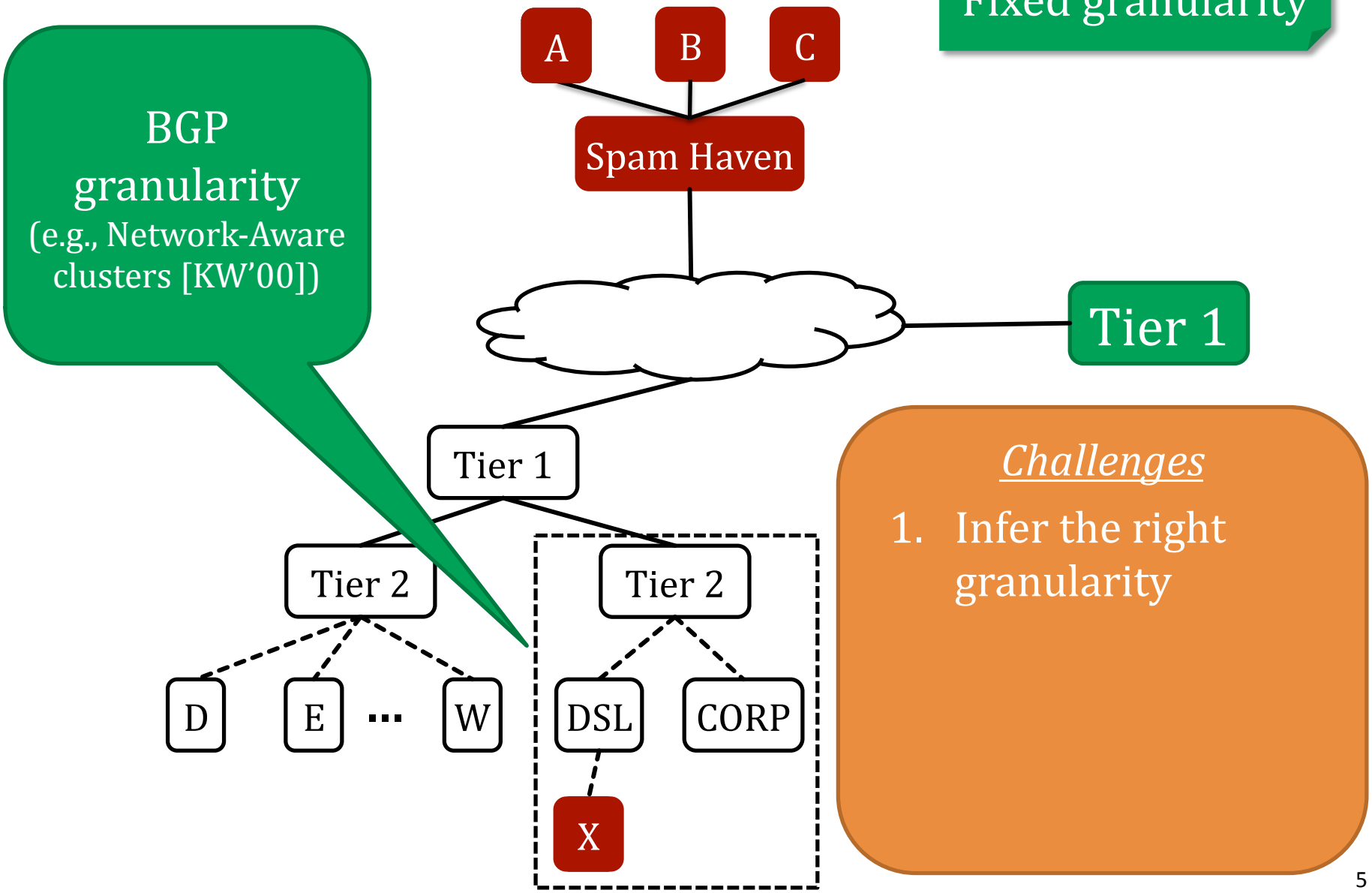
Given a sequence of labeled IP's

1. Can we identify the specific regions on the Internet that have *changed* in malice?

2. Are there regions on the Internet that change their malicious activity more frequently than others?

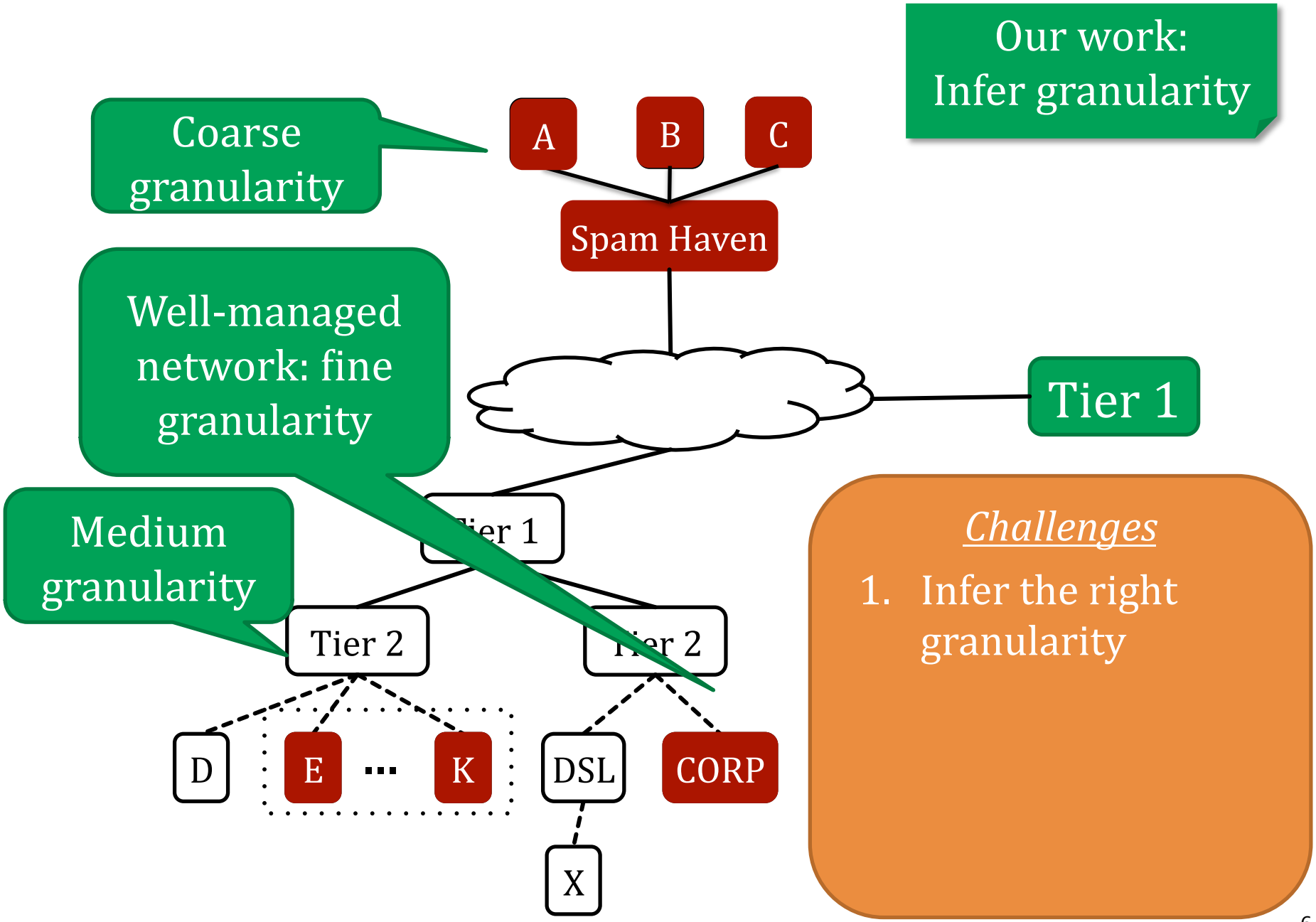

Previous work:
Fixed granularity
Per-IP
Granularity
(e.g., Spamcop)
A
B
C
Spam Haven
Tier 1
Tier 1
Tier 2
Tier 2
D
E
...
K
DSL
CORP
X
Challenges
1. Infer the right granularity

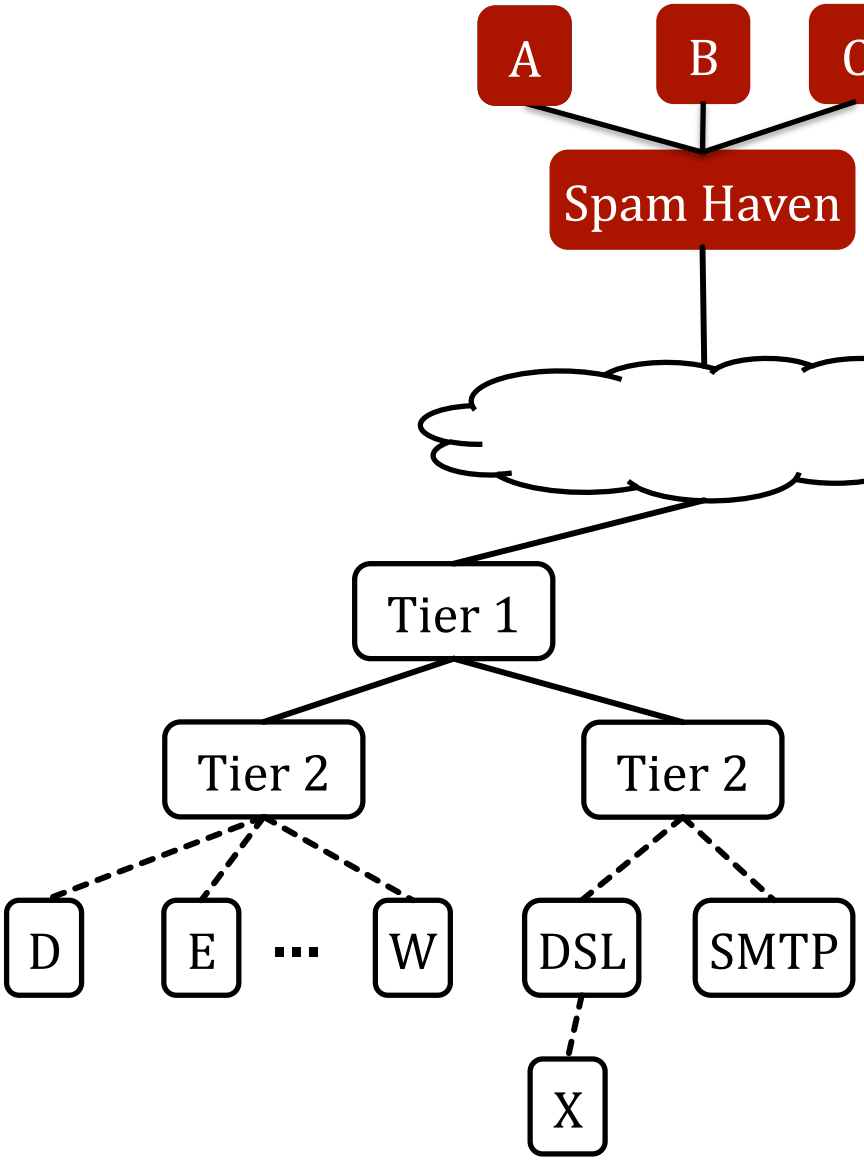
Previous work:
Fixed granularity



Challenges

1. Infer the right granularity


Our work:
Infer granularity
Coarse granularity
A
B
C
Spam Haven
Well-managed network: fine granularity
Tier 1
Medium granularity
Tier 2
Tier 2
D
E
K
DSL
CORP
X
Challenges
1. Infer the right granularity



fixed-memory device
high-speed link

Challenges

1. Infer the right granularity
2. We need *online* algorithms

Research Questions

Given a sequence of labeled IP's

We Present

1. Can we identify the specific regions on the Internet that have *changed* in malice?

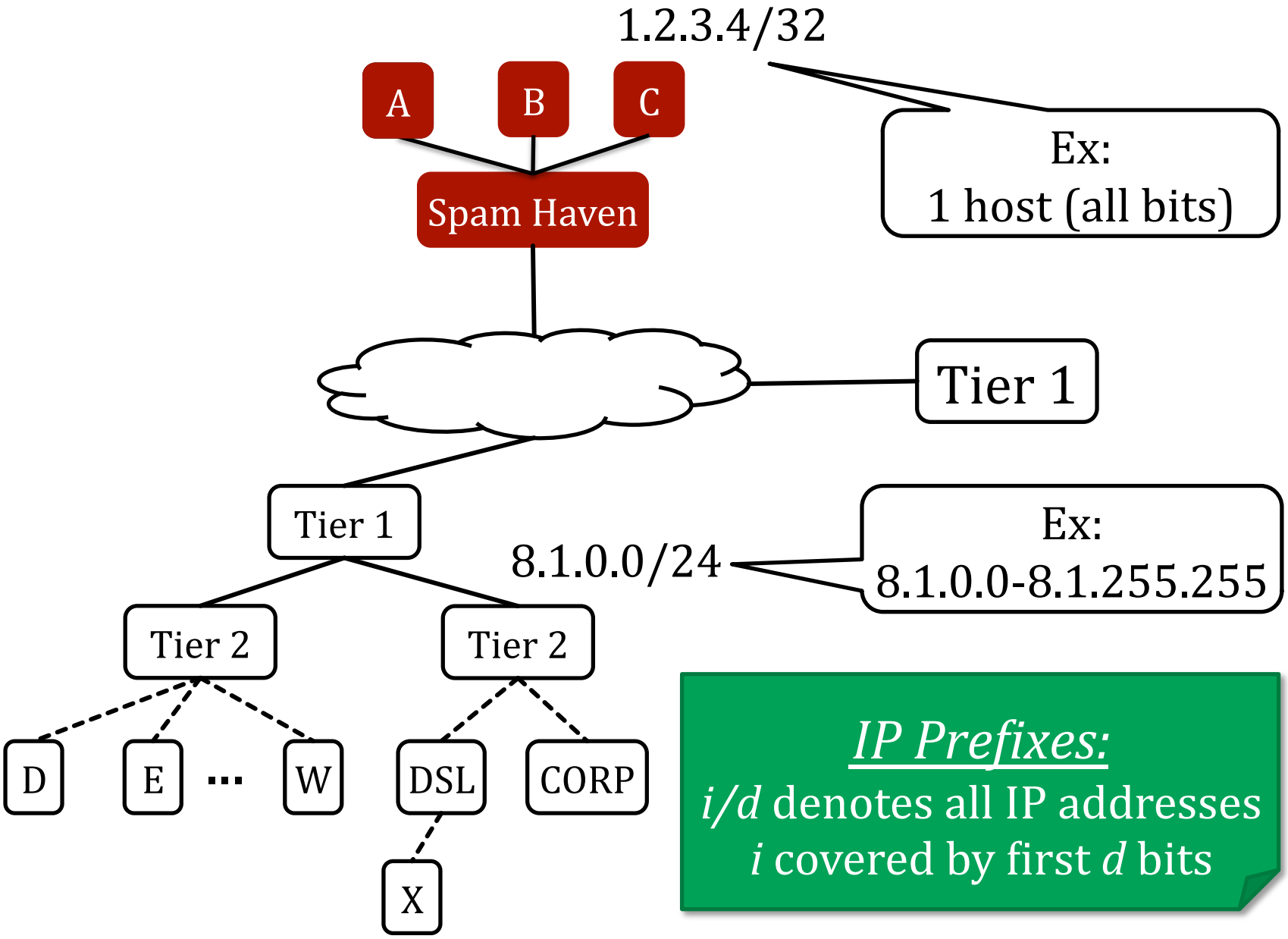
Δ-Change

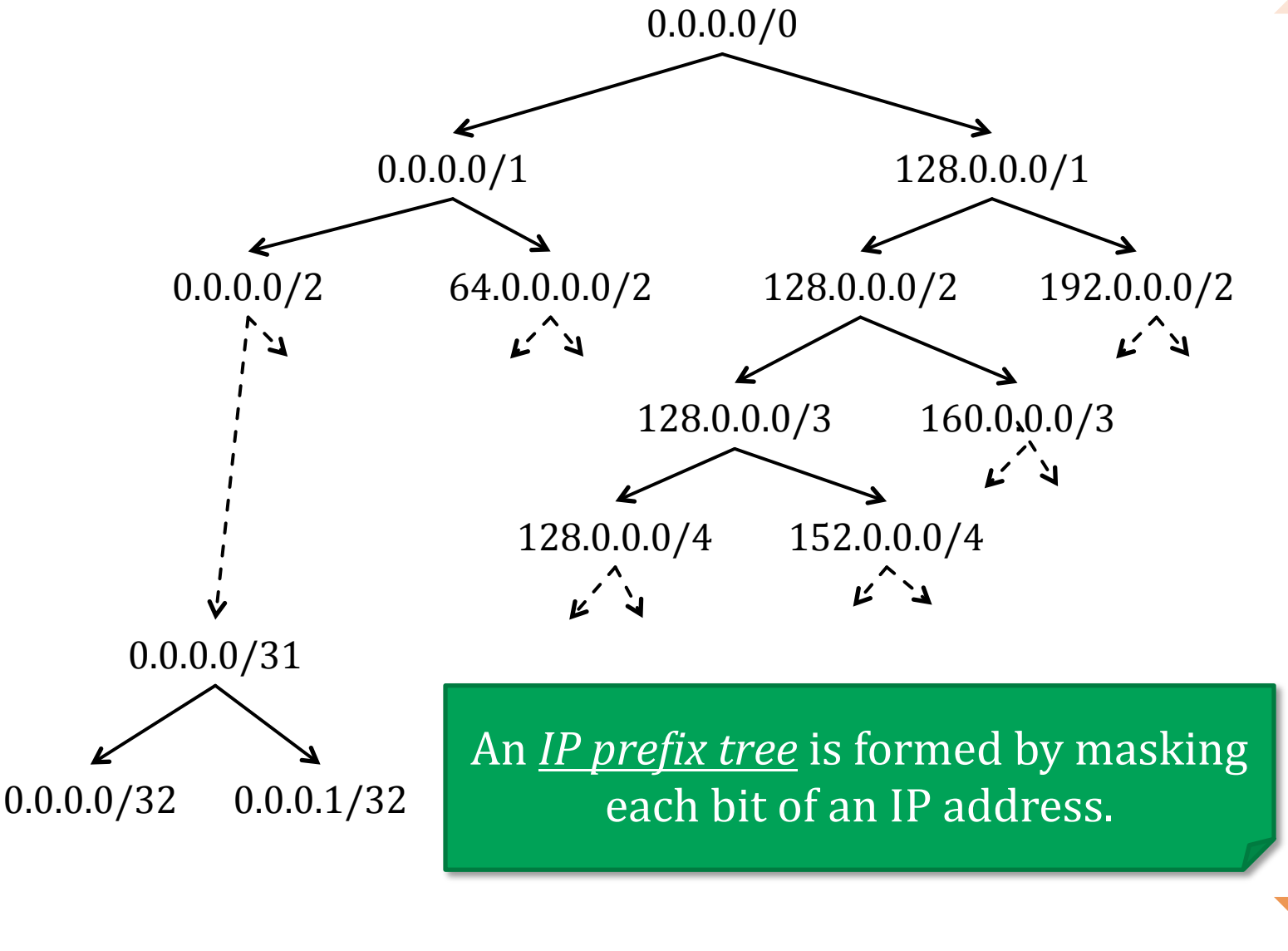
2. Are there regions on the Internet that change their malicious activity more frequently than others?

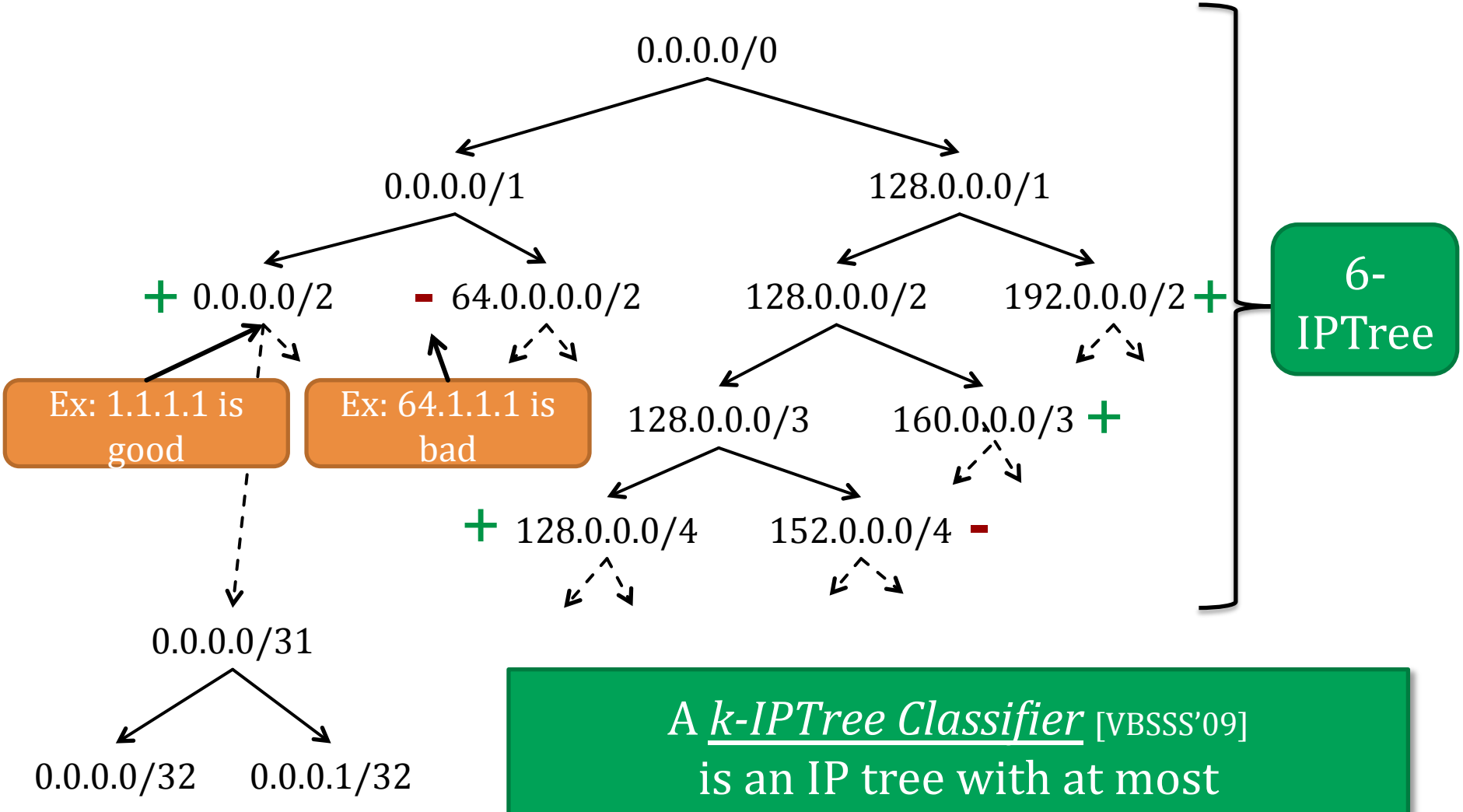
Δ-Motion

Background

1. IP Prefix trees
2. TrackIPTree Algorithm


1.2.3.4/32
A
B
C
Spam Haven
Ex:
1 host (all bits)
Tier 1
Tier 1
8.1.0.0/24
Ex:
8.1.0.0-8.1.255.255
Tier 2
Tier 2
D
E
…
W
DSL
CORP
X
IP Prefixes:
i/d denotes all IP addresses
i covered by first d bits


0.0.0.0/0
0.0.0.0/1
128.0.0.0/1
0.0.0.0/2
64.0.0.0/2
128.0.0.0/2
192.0.0.0/2
128.0.0.0/3
160.0.0.0/3
128.0.0.0/4
152.0.0.0/4
0.0.0.0/31
0.0.0.0/32
0.0.0.1/32
Whole Net
One Host
An IP prefix tree is formed by masking each bit of an IP address.



A *k-IPTree Classifier* [VBSSS'09] is an IP tree with at most k-leaves, each leaf labeled with good ("+) or bad ("-").

TrackIPTree Algorithm

[VBSSS'09]

In: stream of labeled IPs

... <ip_4,+> <ip_3,+> <ip_2,+> <ip_1,->

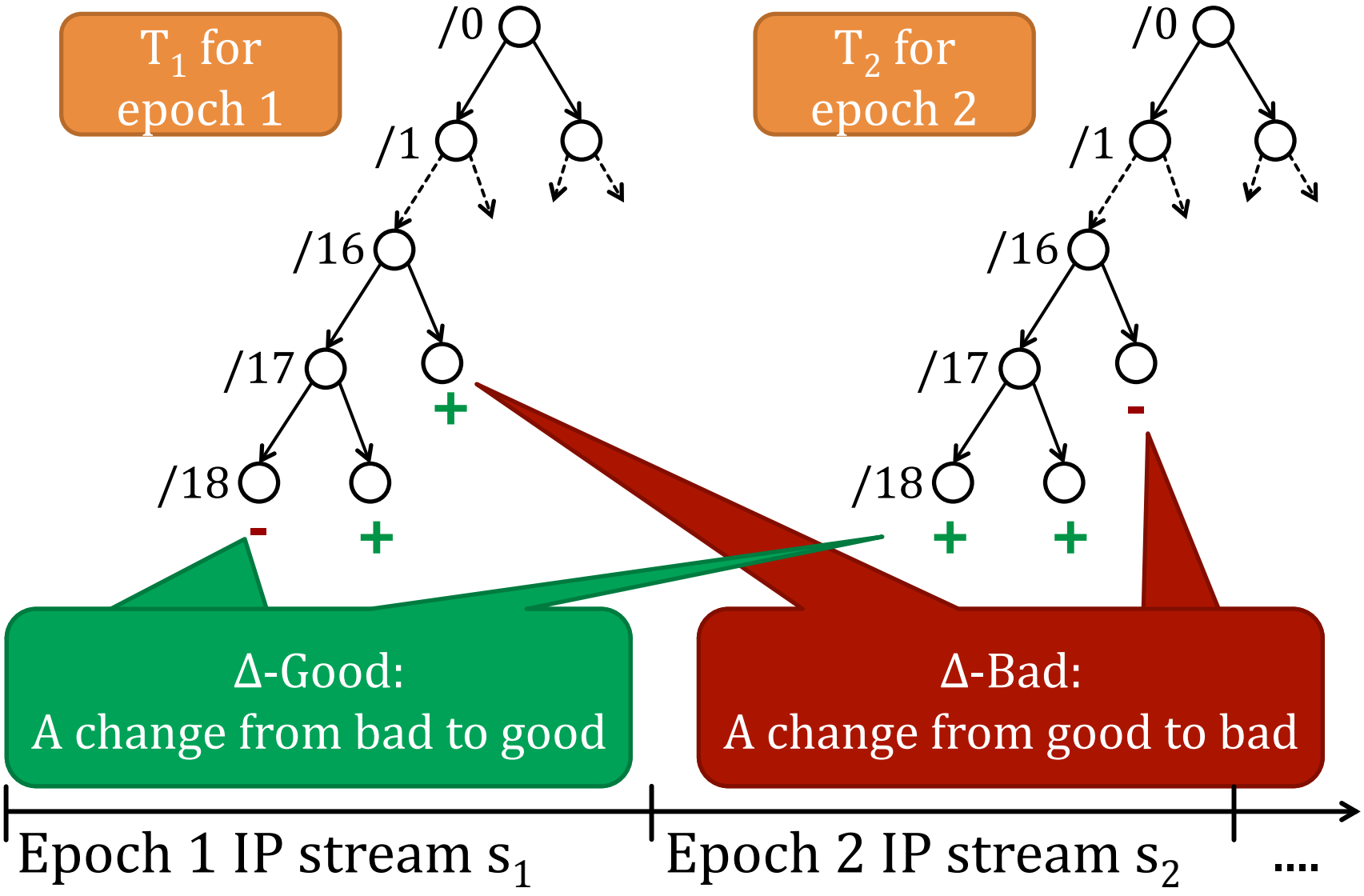
TrackIPTree

Out: k-IPTree

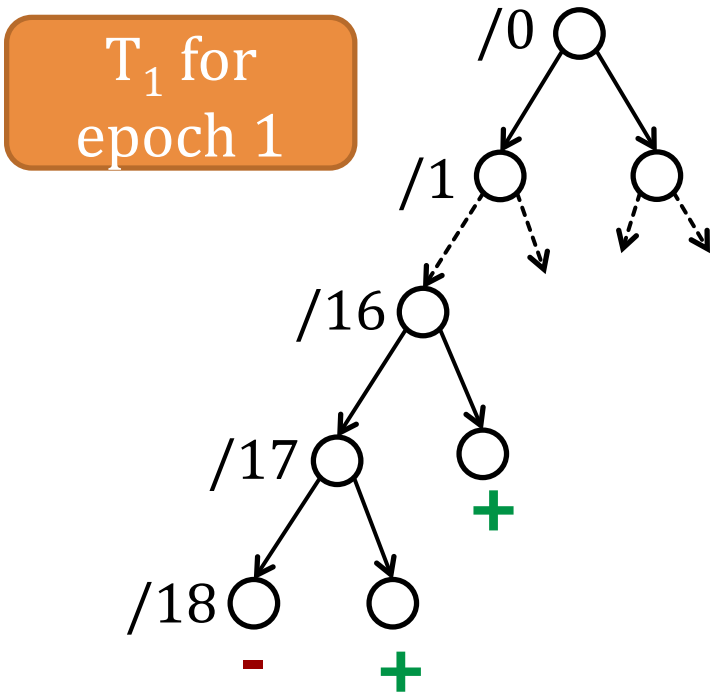
Δ-Change Algorithm

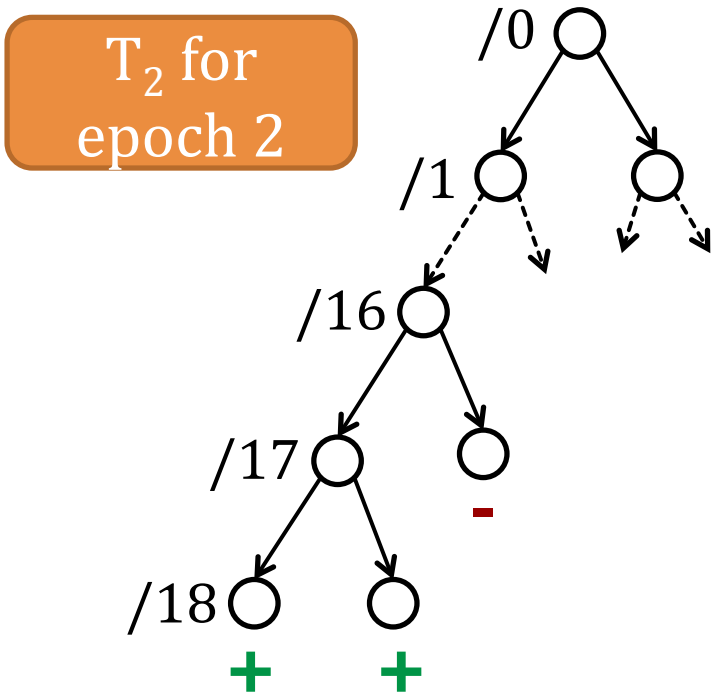
1. Approach
2. What doesn't work
3. Intuition
4. Our algorithm

Goal: identify online the specific regions on the Internet that have changed in malice.



Goal: identify online the specific regions on the Internet that have changed in malice.

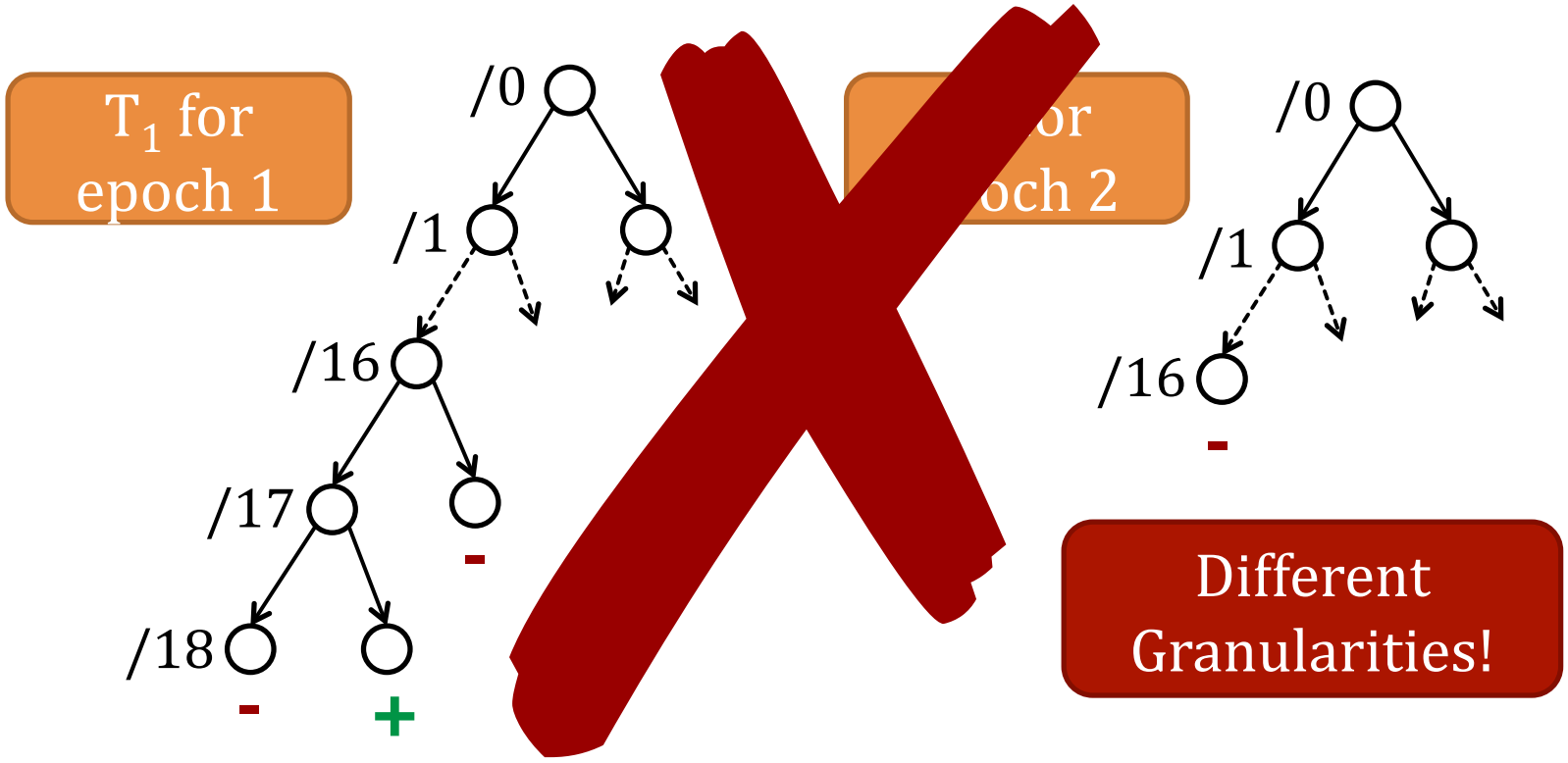




False positive: Misreporting that a change occurred

False Negative: Missing a real change

Goal: identify online the specific regions on the Internet that have changed in malice.



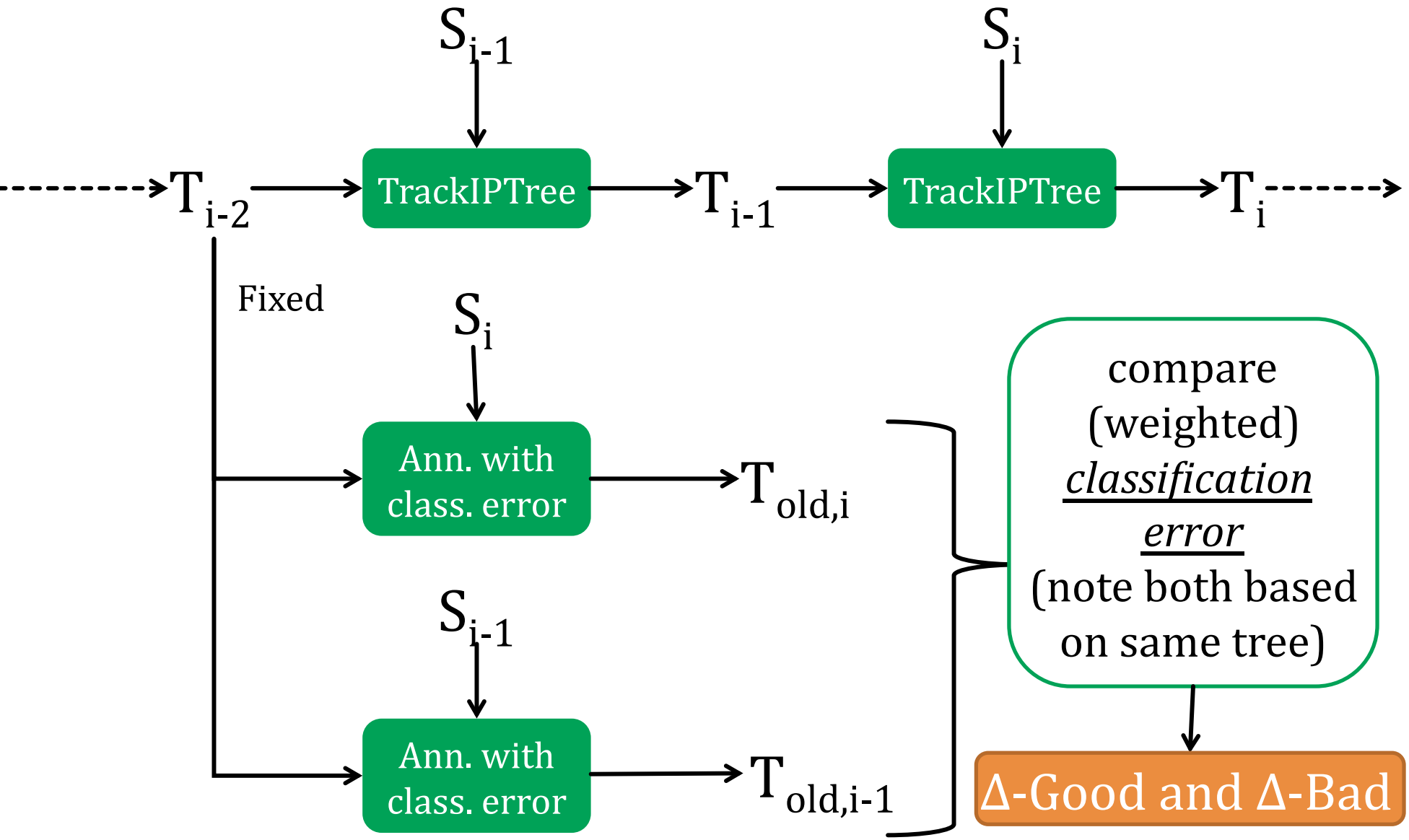
Idea: divide time into epochs and diff

- Use TrackIPTree on labeled IP stream s_1 to learn T_1
- Use TrackIPTree on labeled IP stream s_2 to learn T_2
- Diff T_1 and T_2 to find Δ-Good and Δ-Bad

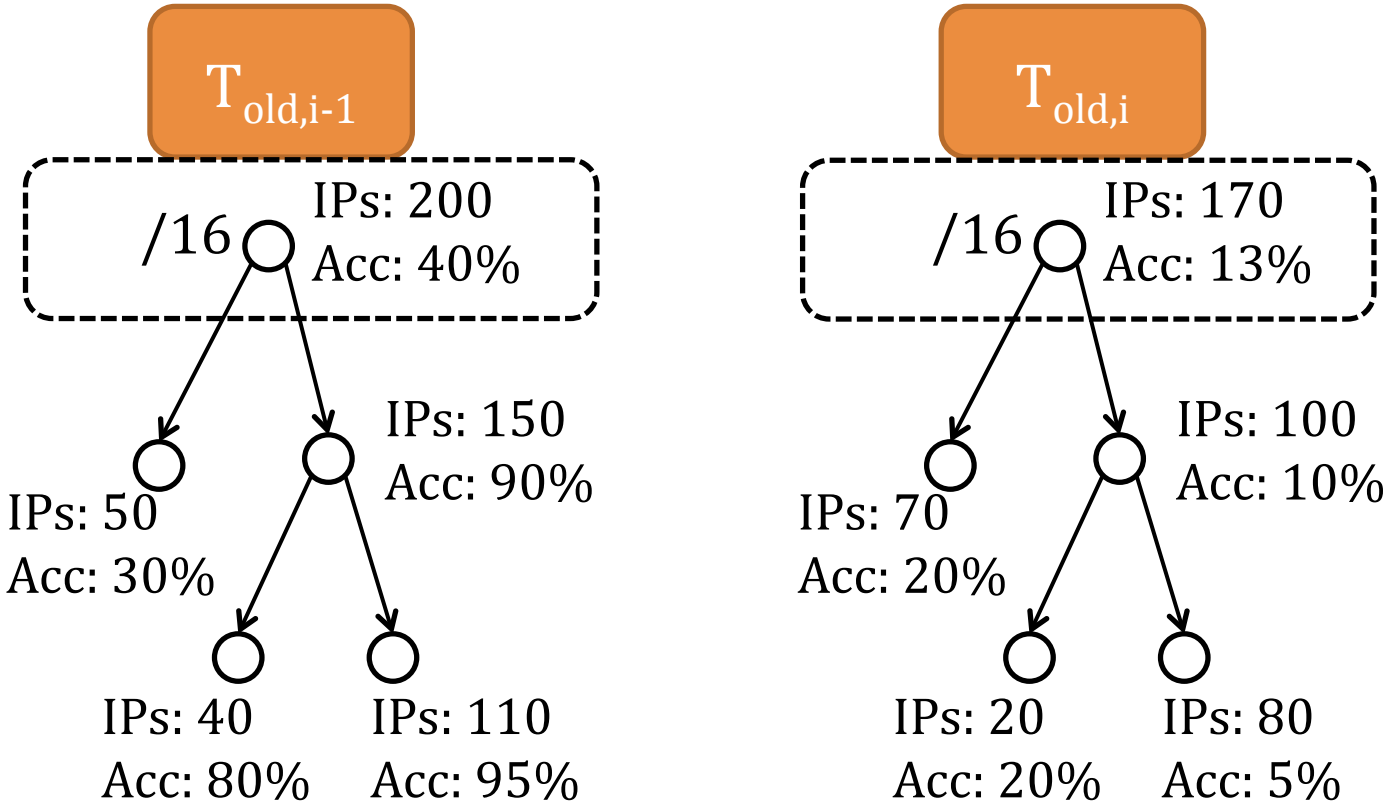
Goal: identify online the specific regions on the Internet that have changed in malice.

Δ-Change Algorithm Main Idea:
Use *classification errors* between T_{i-1} and T_i to infer Δ-Good and Δ-Bad

Δ-Change Algorithm

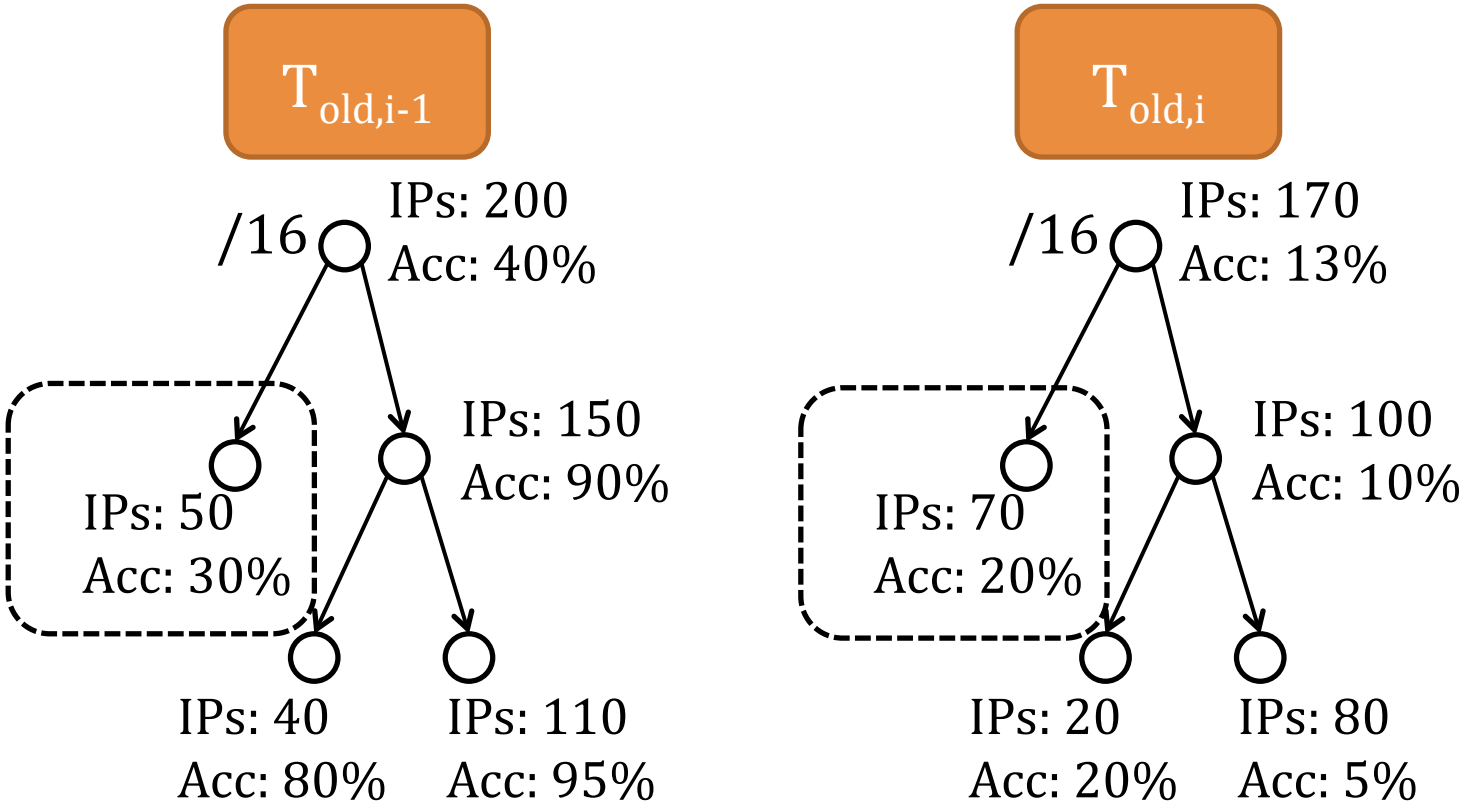


Comparing (Weighted) Classification Error



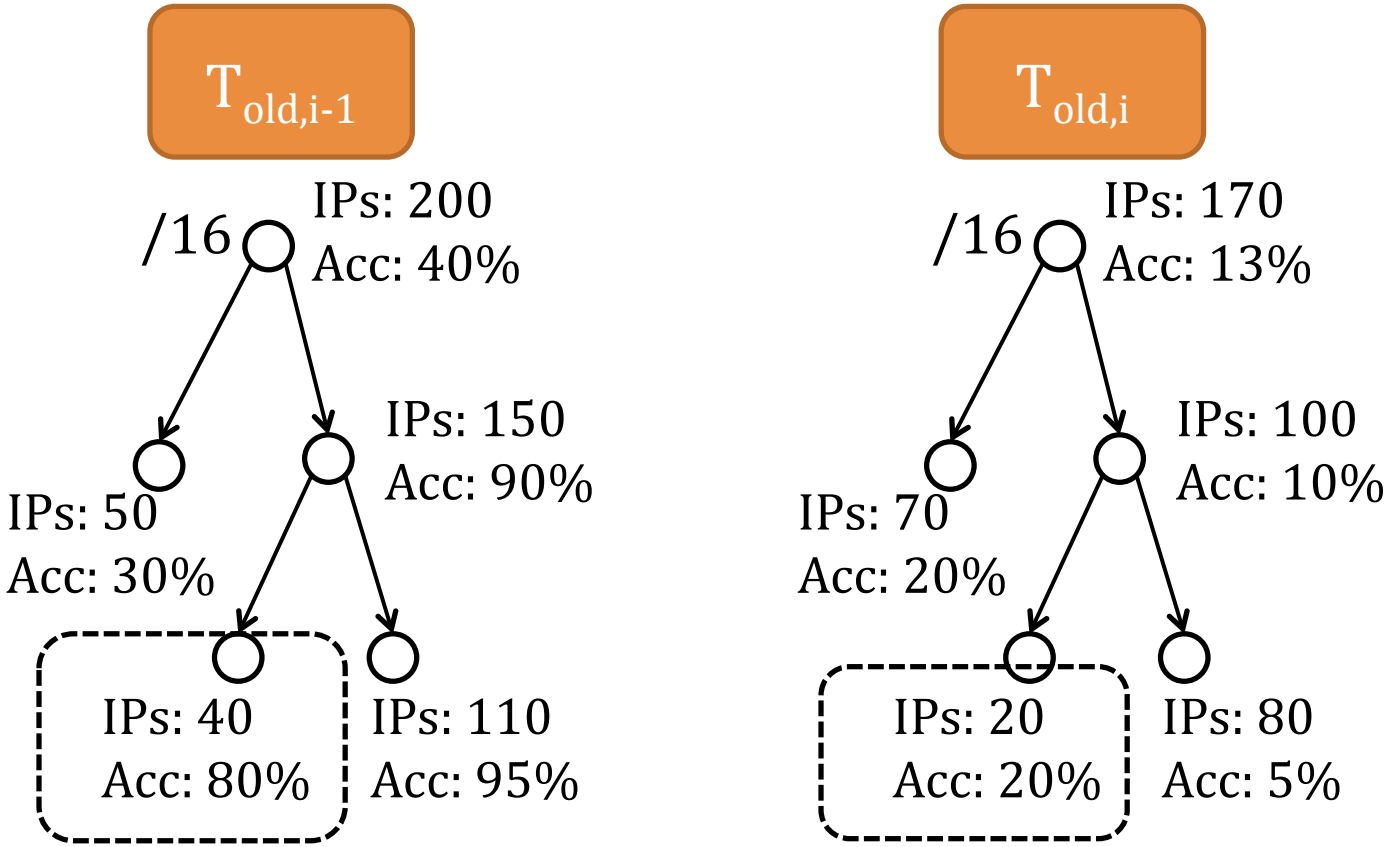
Δ-Change Somewhere

Comparing (Weighted) Classification Error



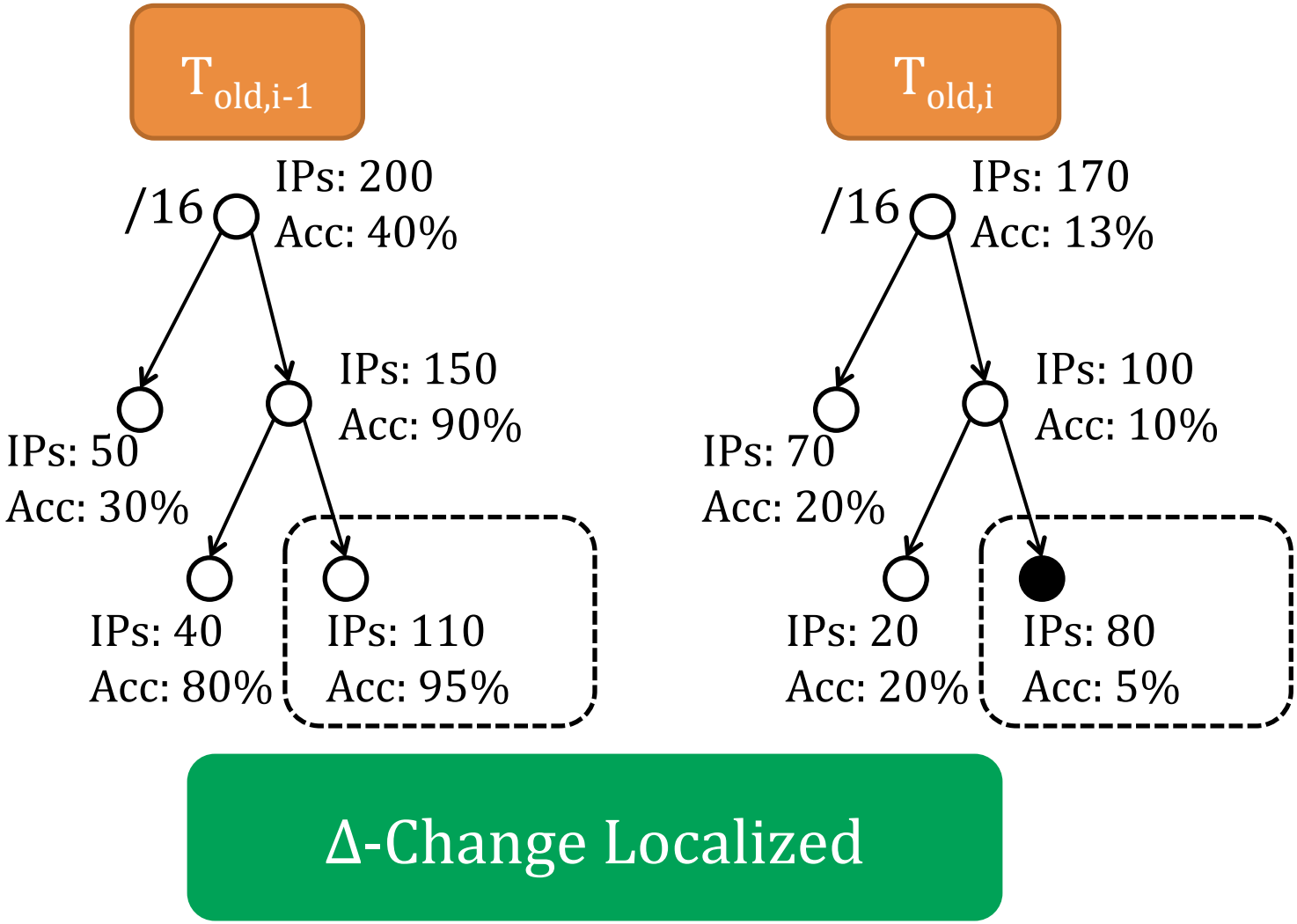
Insufficient *Change*

Comparing (Weighted) Classification Error



Insufficient *Traffic*

Comparing (Weighted) Classification Error



Evaluation

1. What are the performance characteristics?
2. Are we better than previous work?
3. Do we find cool things?

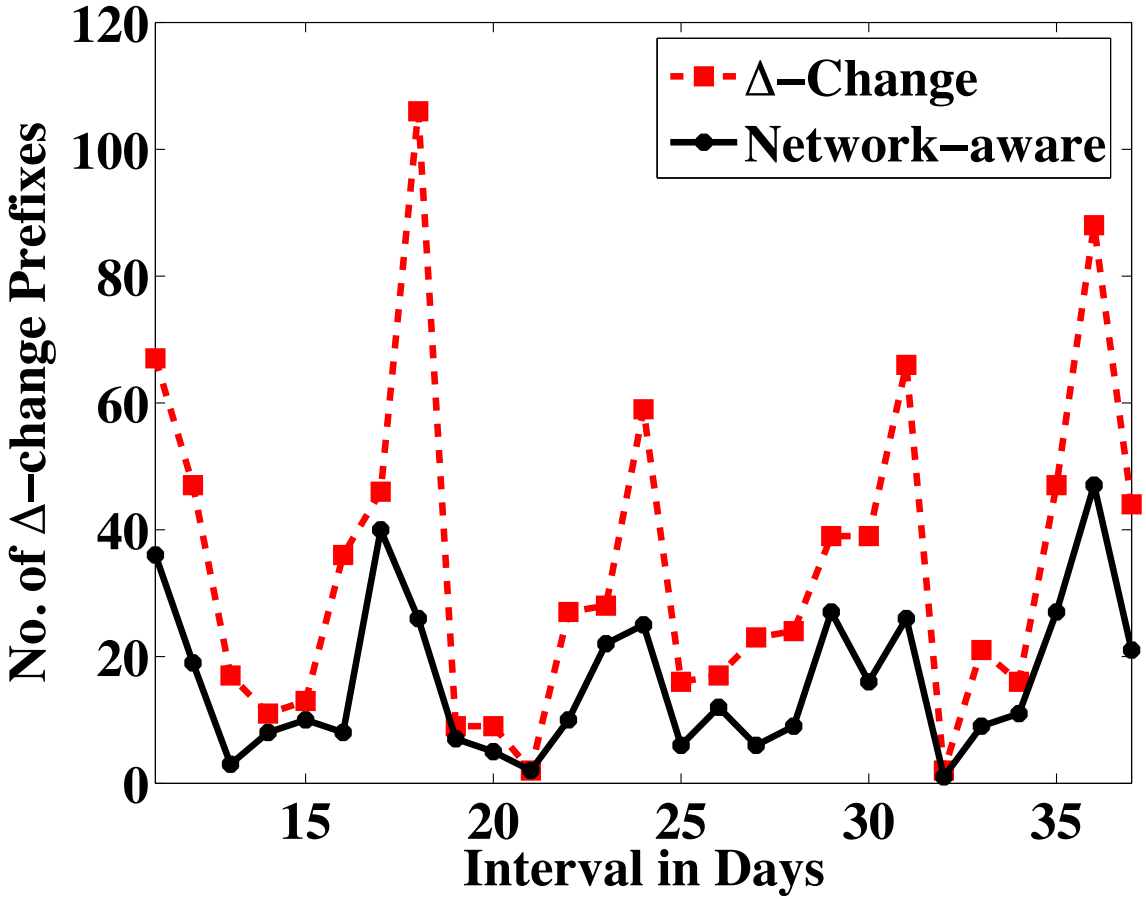
Performance

In our experiments, we :

- let k=100,000 (k-IPTree size)
- processed 30-35 million IPs (on day's traffic)
- using a 2.4 Ghz Processor

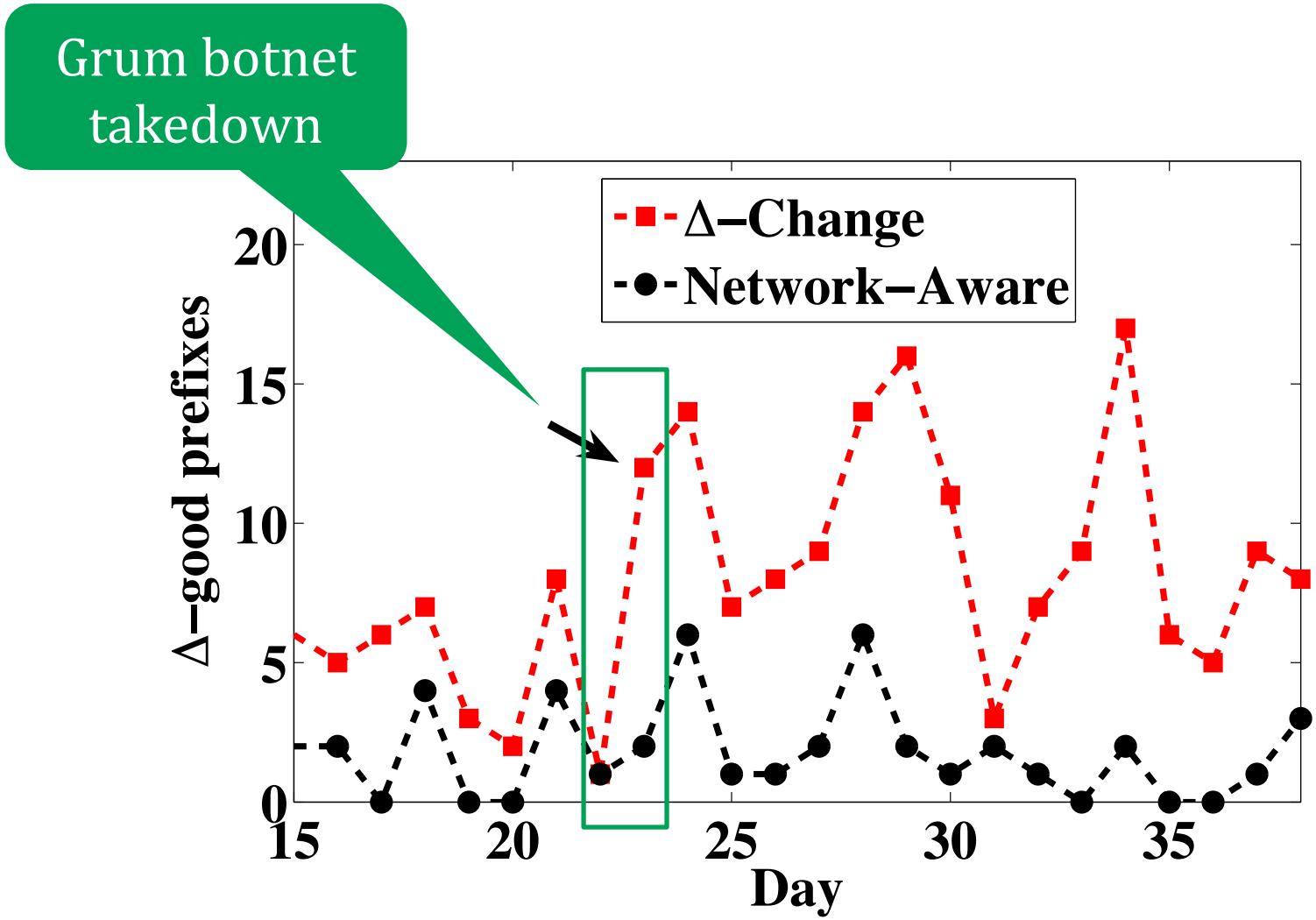
Identified Δ-Good and Δ-Bad
in <22 min using <3MB memory

How do we compare to network-aware clusters? (By Prefix)



2.5x as many changes on average!

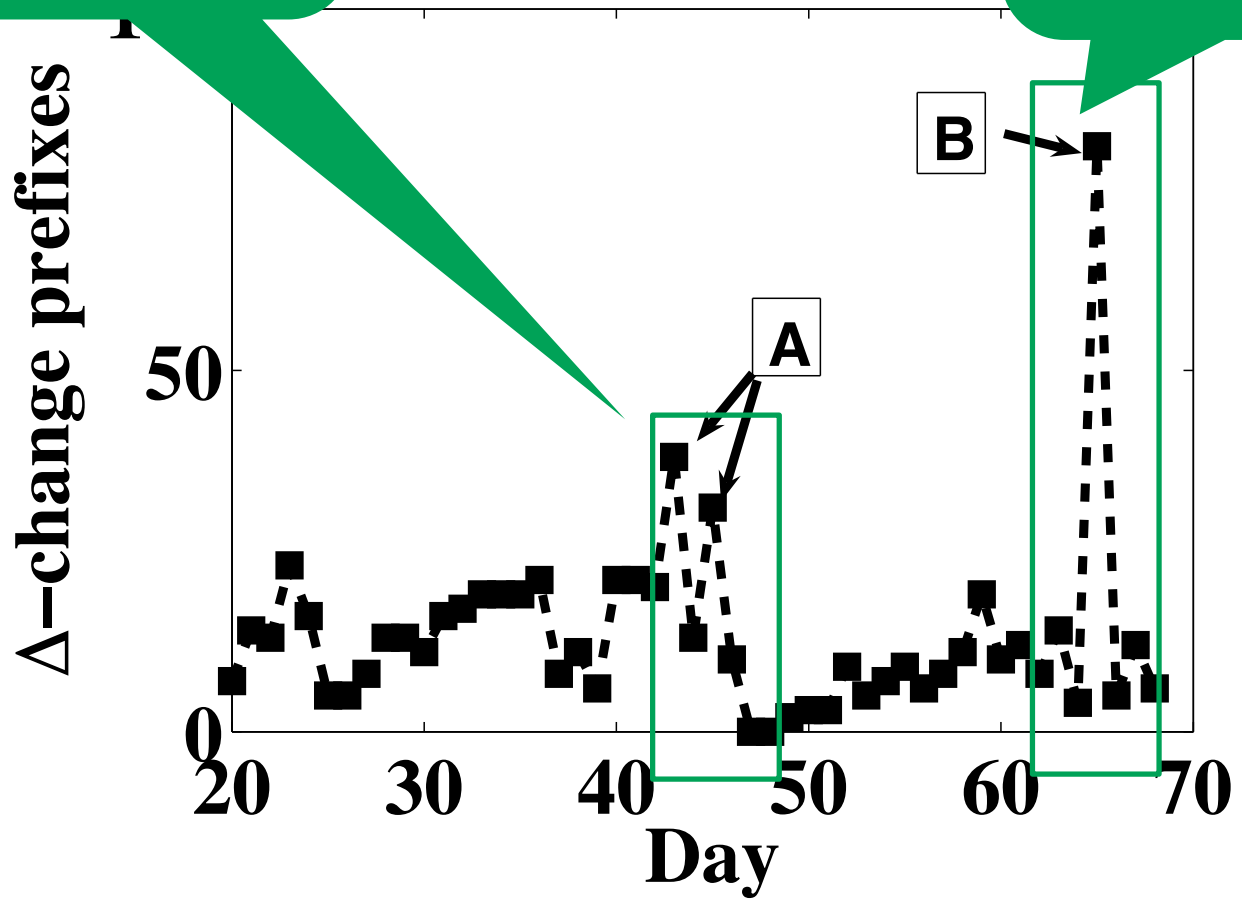
Spam


Grum botnet takedown
Δ–Change
Network–Aware
Δ–good prefixes
0
5
10
15
20
15
20
25
30
35
Day

Botnets

22.1 and 28.6 thousand new DNSChanger bots appeared

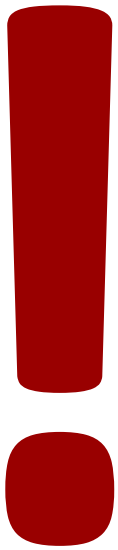
38.6 thousand new Conficker and Sality bots



Caveats and Future Work

"For any distribution on which an ML algorithm works well, there is another on which is works poorly."

– The "No Free Lunch" Theorem



Our algorithm is efficient and works well in practice.

....but a very powerful adversary could fool it into having many false negatives. A formal characterization is future work.

Conclusion

Δ-Change and Δ-Motion: two new *online* algorithms for capturing how malice evolves on the internet

- Scalable
- Discovers right IP granularity
- Finds cool changes


Questions?

END