清華大學
Tsinghua University

Ares: Comprehensive Path Hijacking Detection via Routing Tree

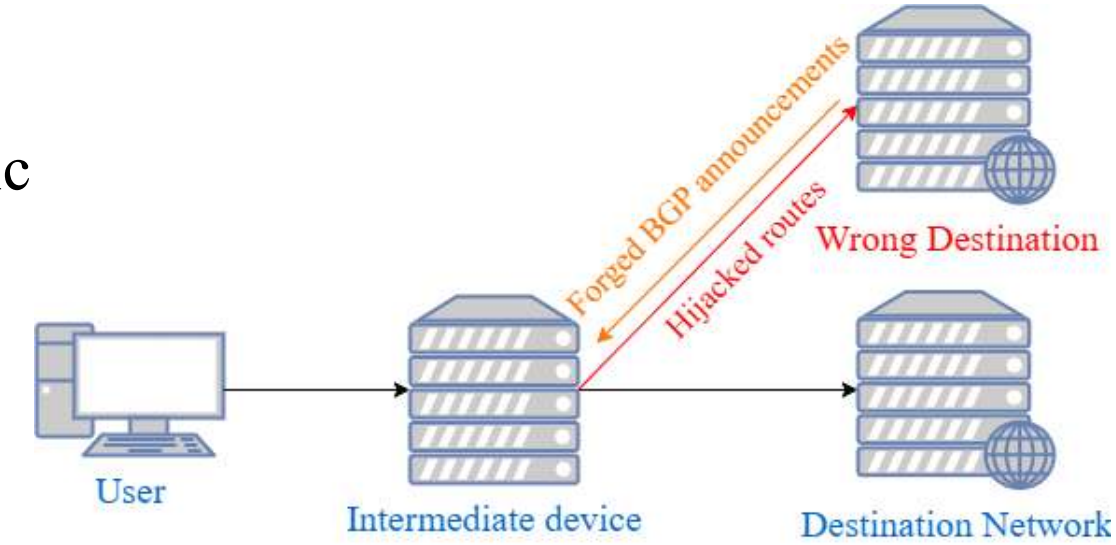
Yinxiang Tao, Chengwan Zhang, Changqing An,
Shuying Zhuang, Jilong Wang, Congcong Miao

tyx23@mails.tsinghua.edu.cn, Tsinghua University

BGP Hijacking

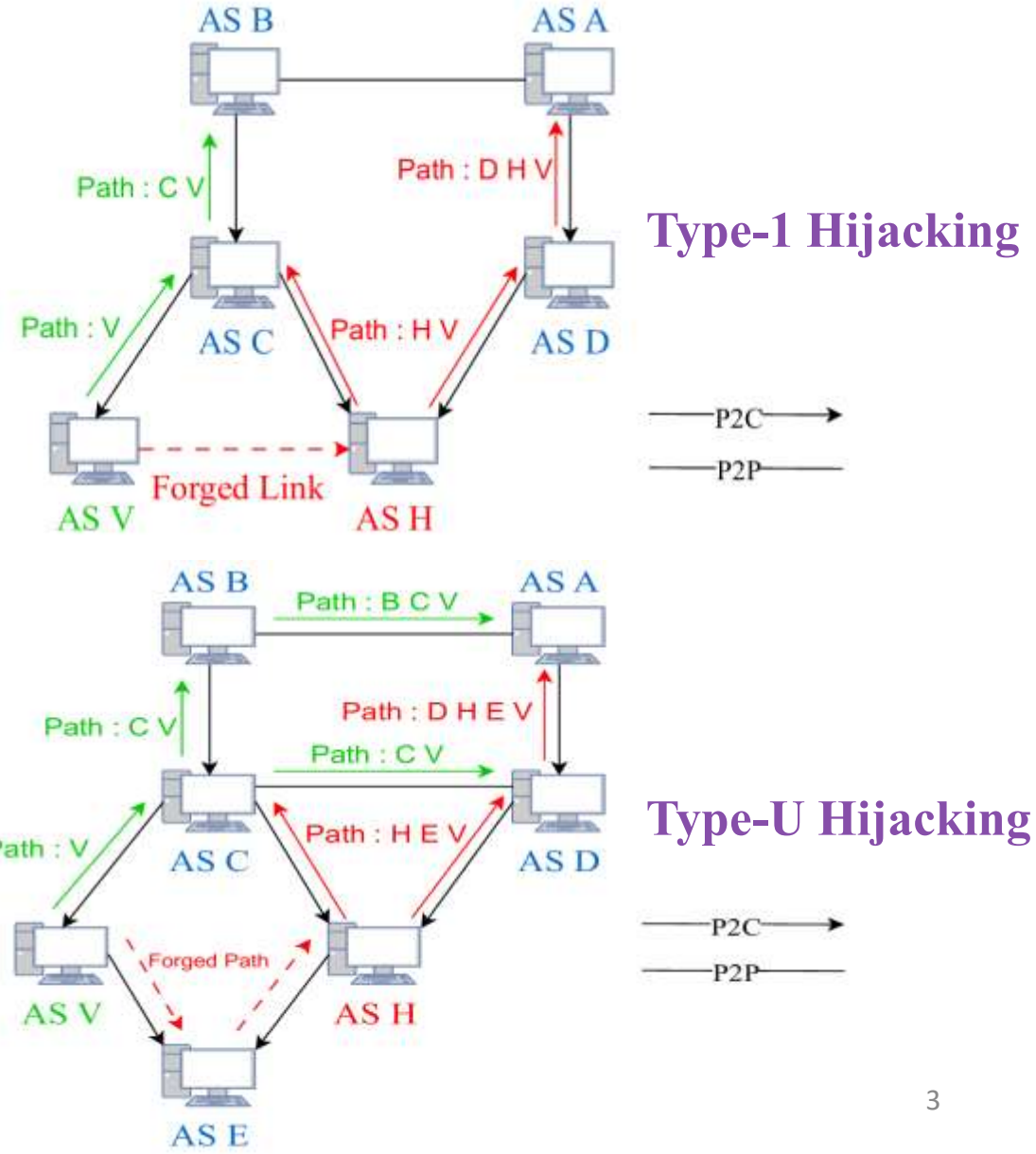
- **BGP Hijacking**: Use forged BGP announcements to redirect Internet traffic

- Threats of BGP Hijacking
  - Poses threats to **routing security**
  - Causes **serious lost**



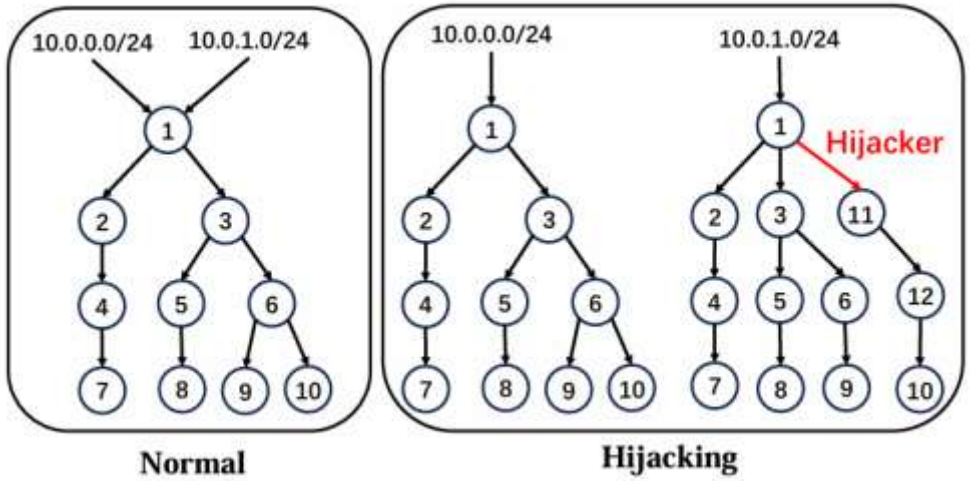
Path Hijacking

- **Origin Hijacking**
  - **Incorrect Origin AS**
  - Well studied and could be detected
  - Could be prevented by methods such as ROV
- **Path Hijacking**:
  - **Correct Origin AS but forged AS paths**
  - Capable of **bypassing ROV**
  - **Type-N (N ≥ 1) hijacking:** Hijacker uses a forged AS path of length N to hijack the victim
  - **Type-U hijacking (Defcon hijacking):** Hijacker prepends a real path to the victim
- Existing Hijack detection methods are not effective against Path hijacking, **especially Defcon hijacking**



What We Do

- Propose a new path hijacking detection method —— **Ares**
  - Key Observation : Hijacking triggers the **creation of a new routing tree**
  - Basic Idea : Detect path hijacking by monitoring **newly emerged routing tree** and compare it with **existing ones** in the same origin AS
- Validate Ares' performance via various data sources
  - Effectiveness (real-world and simulated hijacking events)
  - Efficiency (Runtime overhead in real-world)
- Compared to existing methods
  - Covers all types of path hijacking
  - More efficient
  - Has comparable false positive rate
  - Easy to deploy



A Prefix's Routing Tree consists of best paths to this prefix observed by VPs, which changes when hijacking occur

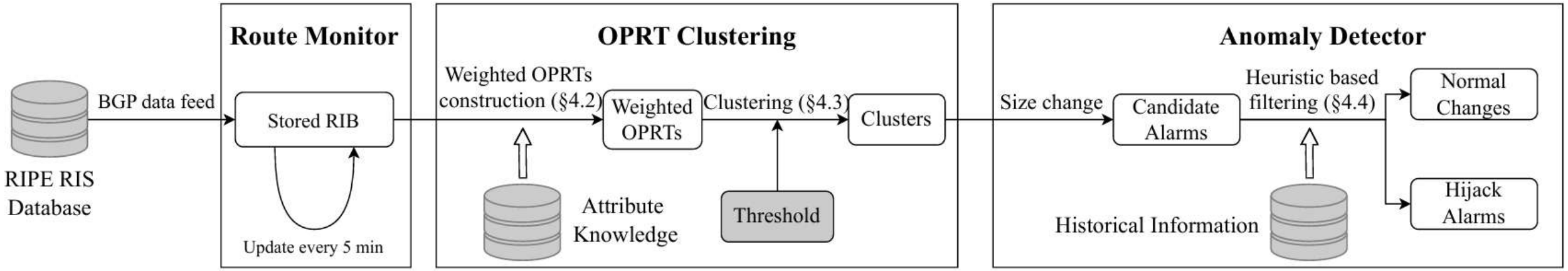
Overall Design

Challenges

1. How to differentiate between normal and hijacked routing trees?
2. Monitoring all routing trees is costly
3. Normal BGP updates also lead to changes of routing trees

Solutions

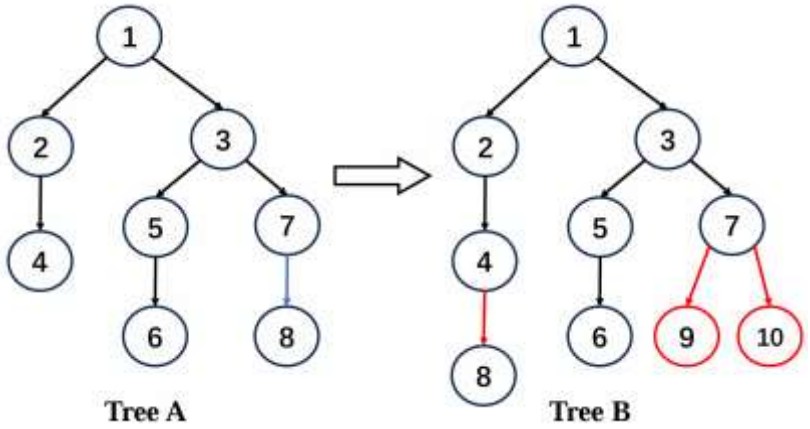
1. Design Weighted Edit Distance to differentiate routing trees
2. Design Clustering algorithm to increase efficiency
3. Design Heuristic Filters to mitigate impact caused by normal BGP updates



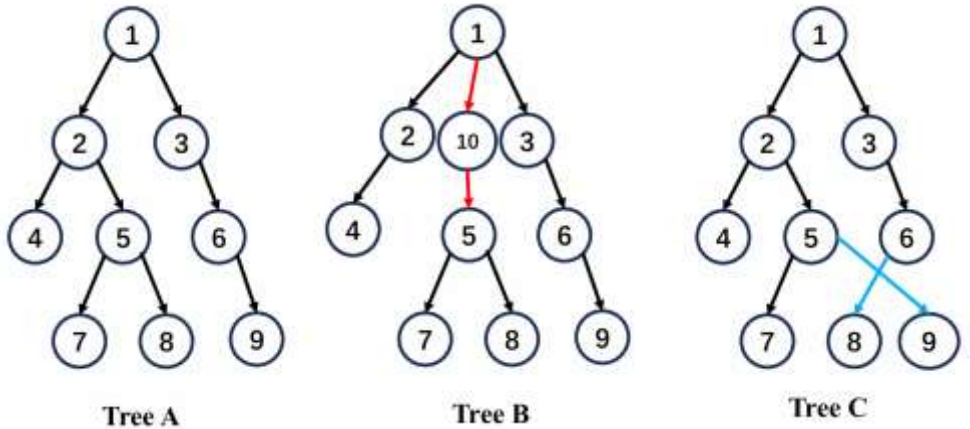
Architecture of Ares

Weighted Edit Distance

- Design Purpose : Differentiate between normal and hijacked routing trees
  - Propose Weighted Edit Distance (WED) to amplify their differences
- Assign high weight to **suspicious links**
  - Links with **low frequency** have high weights (Frequency : Times a link appear in routing table)
  - Links with **large geographic distance** have high weights
- Assign high weight to **impactful links**
  - Links **observed by more VPs** or **nearer to the Origin AS** have high weights
- Enable Ares to **quantify the difference** between newly emerged Trees and existing ones



WED(A, B) = $\omega_A(7,8) + \omega_B(4,8) + \omega_B(7,9) + \omega_B(7,10)$



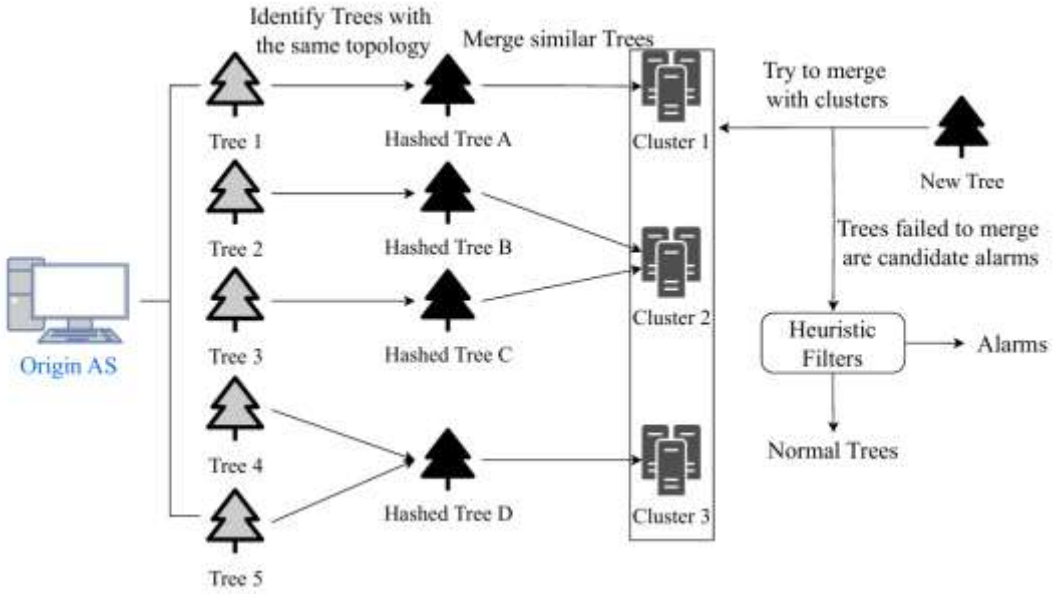
Red links are more impactful than blue links since they are nearer to origin and observed by more VPs

Detection – Clustering

- Detecting hijacking by pairwise WED comparison between routing trees is not feasible
  - High **computational overhead**

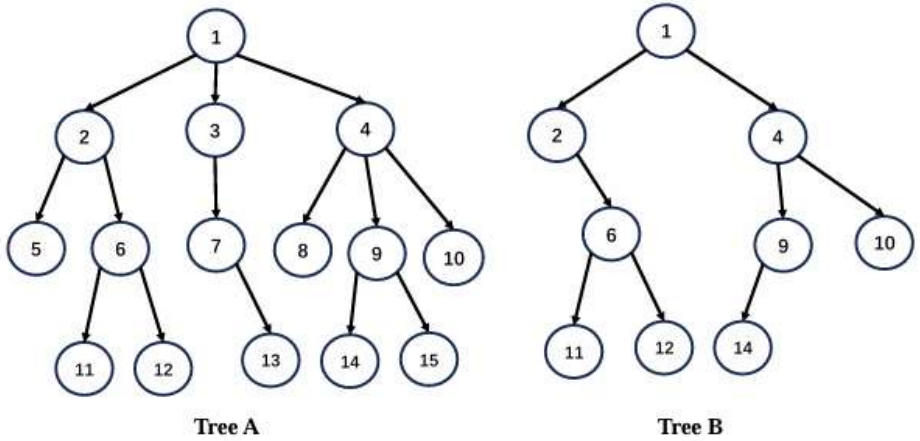
- Design **Clustering** strategy to reduce computational overhead
  1. Merge Trees with the **same topology** (using a hash function)
  2. Merge **similar Trees** into a **Cluster** according to **WED** and **threshold**
  3. After Clustering, only compare **newly emerged Trees** with **Clusters** to detect hijacking

- If a **newly emerged Tree** could not be merged into any **existing Clusters**, it is regarded as **candidate alarm**



Detection – Heuristic Filters

- Design **Heuristic Rules** to mitigate impacts caused by normal BGP updates
  - Mitigate the impact of **incomplete BGP convergence**
  - Capture **typical hijack patterns**
  - Leverage **historical routing context**
- Heuristic I – Partial Edit Distance
  - The **disappearance** of edges does not indicate a hijack
- Heuristic II – Size of new cluster
  - A hijacker usually attacks **few prefixes** instead of many



- Heuristic III – Historical information
  - Compare the routing trees in an AS with their **former versions** if it cannot be merged
- Heuristic IV – Existing relationships
  - Find links with **large weights** and check whether they have **business relationship** or **violate valley-free**

Evaluation – Historical Events

- Successfully detected **all** historical path hijacking events with **low alarm numbers**
  - **100% recall; 2.31 alarms/h** on average

Event	Date	Detected	Alarm (Confirmed by report)	Duration
trog_1	2014-09-26	√	4(1)	8 hours
trog_2	2014-09-26	√	4(1)	8 hours
trog_3	2014-09-27	√	9(1)	8 hours
backconnect_3	2016-02-20	√	24(1)	8 hours
backconnect_5	2016-02-21	√	5(2)	16 hours
backconnect_6	2016-04-16	√	18(2)	8 hours
france_1	2012-12-28	√	8(6)	8 hours
enzu_1	2015-03-26	√	288(257)	8 hours
defcon_1	2008-08-10	√	27(2)	28 hours
facebook_1	2011-03-22	√	31(1)	8 hours
zayo_1	2022-02-03	√	16(1)	8 hours
amazon_1	2022-08-17	√	19(1)	8 hours

Evaluation – Simulated Hijacking Events

- Generated hijacking events on **16 scenarios** via **simulation**
  - Four types of ASes: Tier-1, Content, Enterprise and Transit/Access for both **Hijackers and Victims**
- **High recall** in important hijacking scenarios
  - Exact prefix hijacking / Tier-1 & Content Victims / non-Tier-1 Hijackers : **97.2% recall on average**
  - Sub-prefix hijacking /Tier-1 & Content Victims / non-Tier-1 Hijackers : **99.3% recall on average**

Victim Type	Tier-1		Content	
Hijacker Type	Non-Tier-1	Non-Tier-1	Non-Tier-1	Non-Tier-1
Prefix Type	Exact Prefix	Sub-Prefix	Exact Prefix	Sub-Prefix
Type-1 Hijacking	99.4%	99.4%	96.3%	98.1%
Type-2 Hijacking	98.3%	99.4%	96.8%	98.7%
Type-3 Hijacking	95.3%	100%	96.8%	100%
Type-U Hijacking	98.6%	100%	96.9%	98.8%
All	98.2%	99.7%	96.7%	98.9%

Evaluation – Simulated Hijacking Events

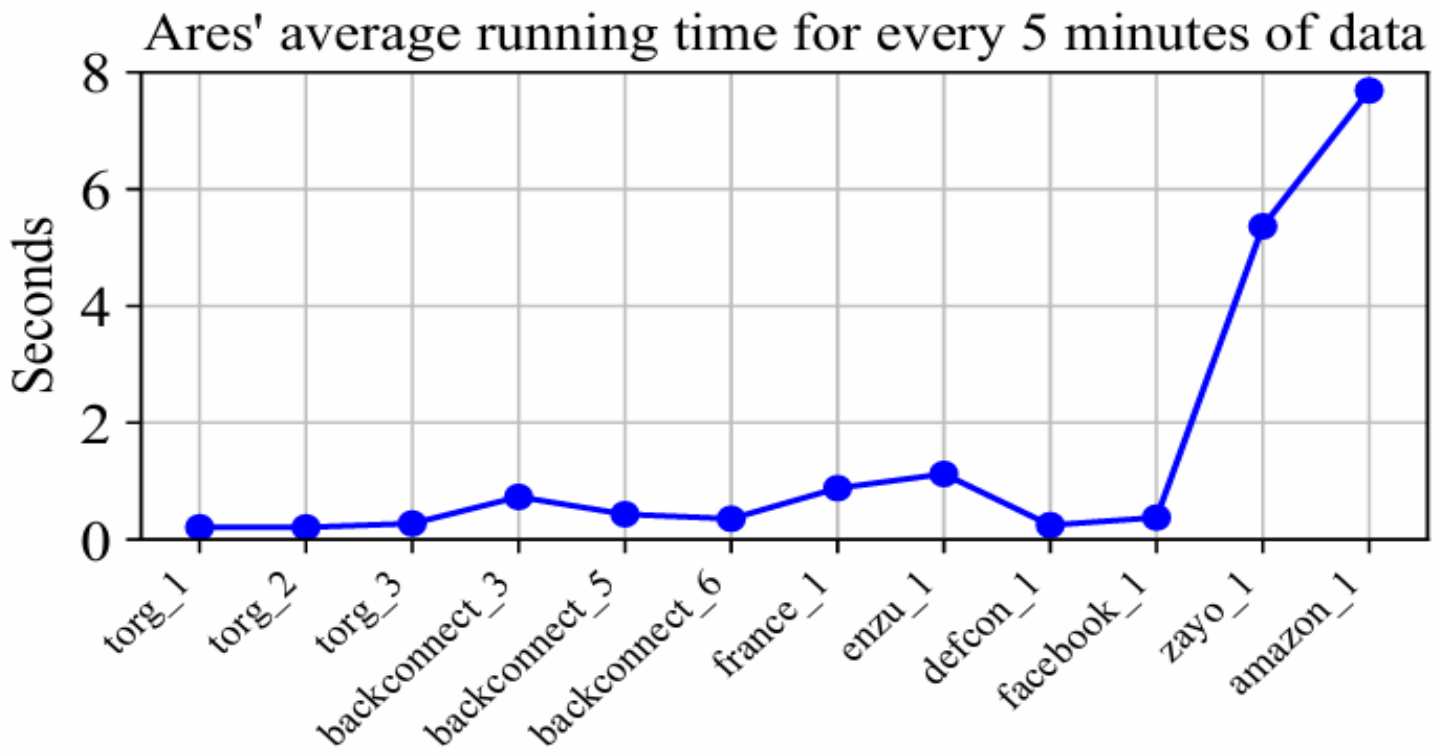
- **Remarkable recall** in all scenarios with non-Tier-1 Hijackers
  - Exact prefix hijacking / Enterprise & T\A / non-Tier-1 Hijackers : **98.5% recall on average**
  - Sub-prefix hijacking /Enterprise & T\A / non-Tier-1 Hijackers : **97.0% recall on average**

Victim Type	Enterprise		Transit/Access	
Hijacker Type	Non-Tier-1	Non-Tier-1	Non-Tier-1	Non-Tier-1
Prefix Type	Exact Prefix	Sub-Prefix	Exact Prefix	Sub-Prefix
Type-1 Hijacking	98.9%	95.6%	99.3%	98.4%
Type-2 Hijacking	97.7%	94.9%	99.6%	99.0%
Type-3 Hijacking	96.6%	94.9%	98.2%	98.4%
Type-U Hijacking	98.3%	95.7%	99.5%	98.8%
All	97.9%	95.2%	99.2%	98.6%

Evaluation – Other Perspectives

- **Low False Positive Rate** against normal BGP changes
  - Sampled BGP changes in real-world
  - **1.06%** False Positive Rate (18/1706)

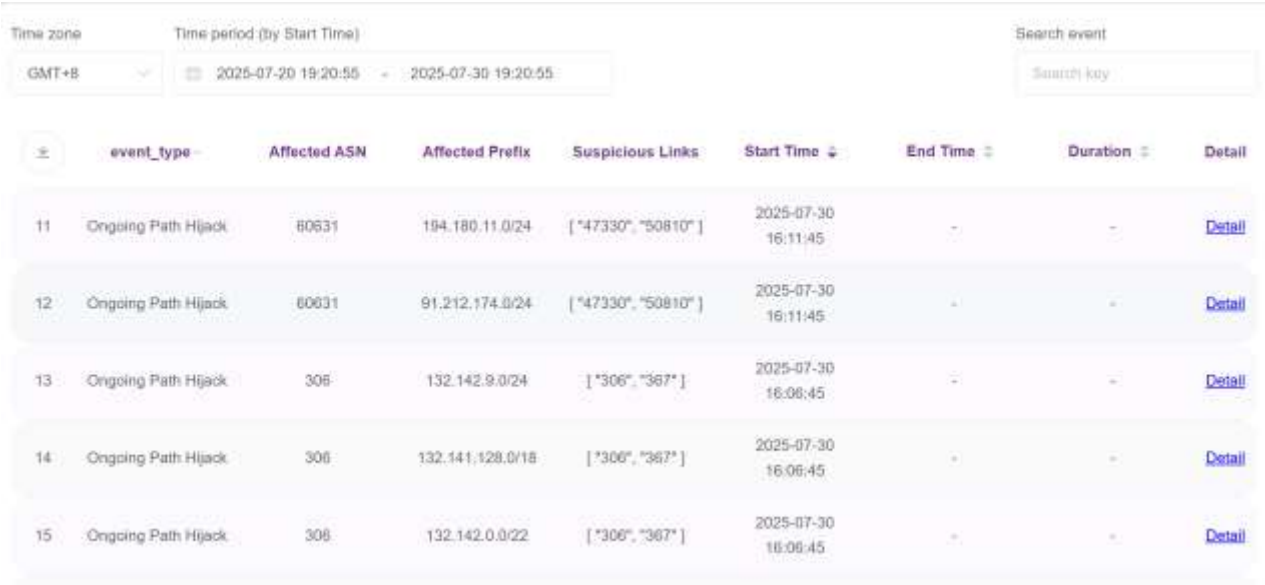
- **Low runtime overhead**
  - High efficiency, capable of **real-time detection**.
  - 6.33s to detect updates in every 5 minutes of a RIPE RIS collector on average at 2024/10



Conclusion

- **Propose Ares: a novel method for path hijacking detection**
  - Covers all types of path hijacking
  - High recall rate and low false positive rate
  - Low runtime overhead, capable of real-time detection

- **Future work**
  - Now integrated in BGPwatch (https://bgpwatch.cgtf.net/#/ares)
  - Will be adjusted to better adapt to real-world applications in the future



BGPWatch : a Collaborative BGP Routing Analyzing and Diagnosing Platform

清華大學
Tsinghua University

Thank You for Listening!

Artifacts : https://doi.org/10.5281/zenodo.15589806

BGPWatch : https://bgpwatch.cgtf.net/