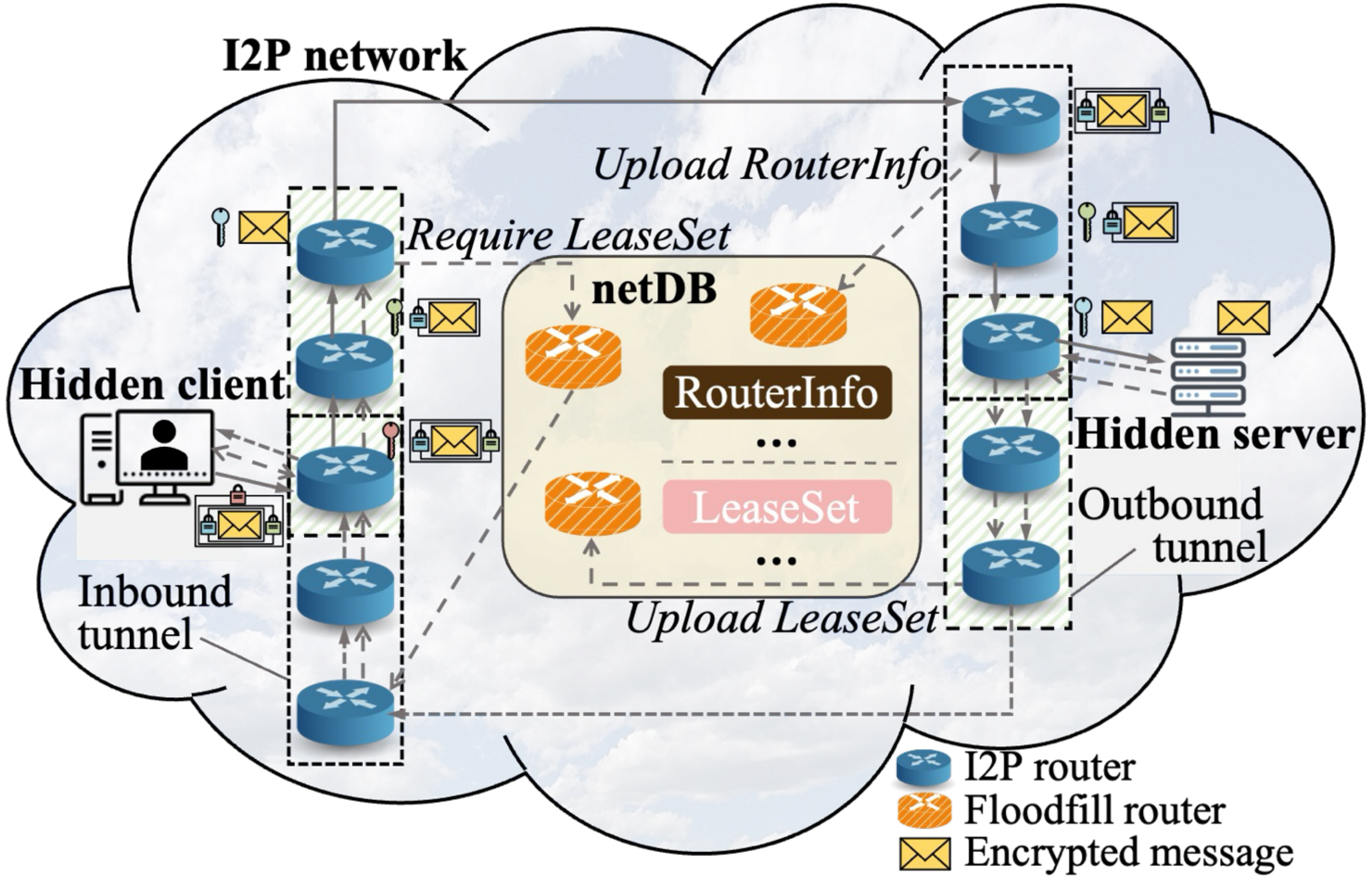
Time will Tell: Large-scale De-anonymization of Hidden I2P Services via Live Behavior Alignment

Hongze Wang[†], Zhen Ling[†], Xiangyu Xu[†], Yumingzhi Pan[†], Guangchi Liu[†], Junzhou Luo[†‡], **Xinwen Fu**[§]

[†] 東南大學 SOUTHEAST UNIVERSITY [‡] 福耀科技大學 FUYAO UNIVERSITY OF SCIENCE AND TECHNOLOGY [§] UMASS University of Massachusetts Lowell

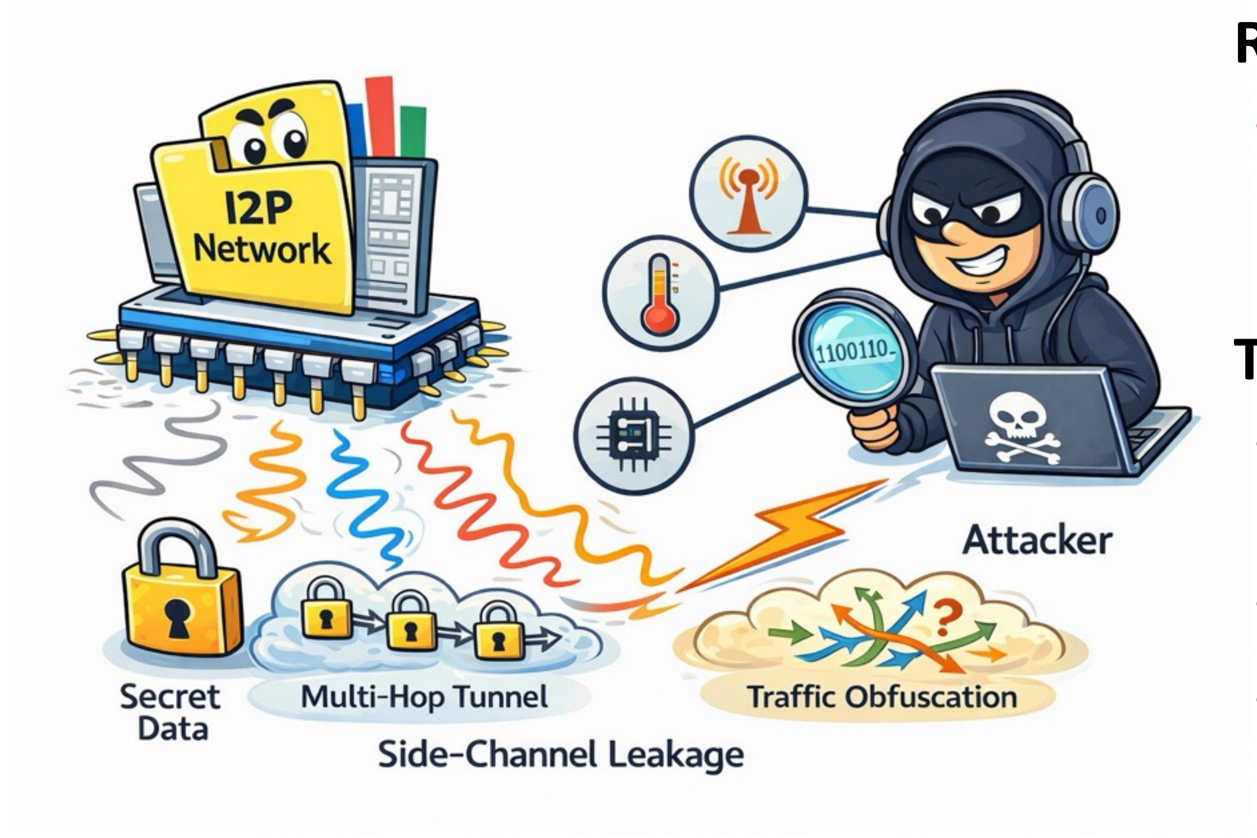
Background: The I2P Network

I2P Architecture and Garlic Routing



Goal & Threat model

Can multi-hop tunnels and traffic obfuscation truly prevent IP deanonymization?



Research Goal

- **Stealthily uncover** the real IP address of a target I2P hidden service.

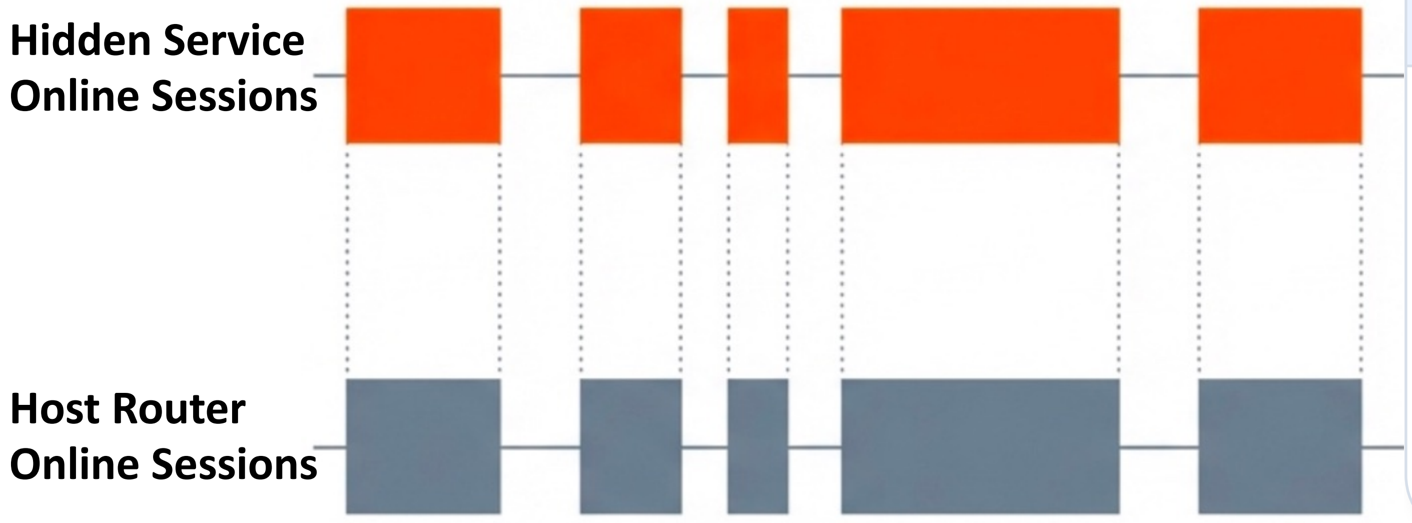
Threat Model & Assumptions

- Deploy **a small number** of floodfill routers to collect RouterInfo traces.
- Target hidden service is **publicly accessible**.
- Target router operates in **non-hidden mode** (default configuration).

Key insight

The core vulnerability: live behavior alignment

If we can infer when each router is online/offline, we can match it with the target service's online/offline pattern.



Why this is dangerous

- User behavior is **inherently diverse**, creating distinctive **temporal fingerprints** across I2P routers.
- We can **bypass** I2P's traffic obfuscation and multi-hop tunnel protections — turning "*time*" into a **side channel**.

From Insight to Practice: Challenges and Solutions

Turning live-behavior alignment into a scalable attack is non-trivial.

! Challenge (C-I) : Large-scale routers' live behavior inference

- Active probing does not scale. (≈45,000 routers/day)
- Active probing risks exposing the attacker's intentions, potentially alerting the targets and prompting evasive actions.
- Passive inference is noisy: RouterInfo publication intervals are randomized.

✓ Solution (S-I): RouterInfo-publication-based Inference

1. Identify **routine** RouterInfo publications.
2. Detect unexpected gaps between routine publications → **coarse-grained behavior**.
3. Use startup/shutdown pattern to identify online session boundaries → **fine-grained behavior**.

Outcome: Passive monitoring → fine-grained online-session inference methods.

From Insight to Practice: Challenges and Solutions

Turning live-behavior alignment into a scalable attack is non-trivial.

! Challenge (C-II) :
Inference Accuracy under Low-cost Strategy

- Limited floodfill deployment → incomplete RouterInfo collection.
- Probabilistic propagation + router churn → inevitable missing publications.
- Missing RouterInfos distort session boundaries.
- Distorted sessions cause behavior-alignment mismatches.

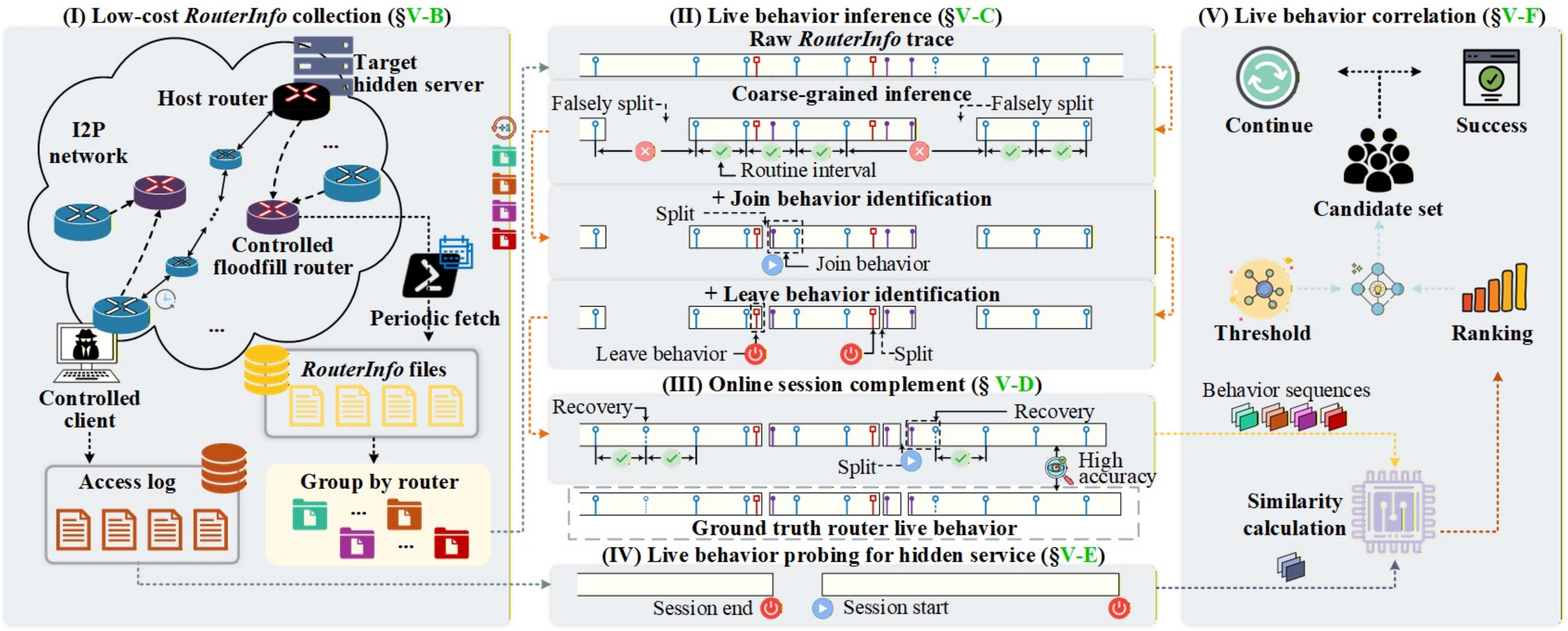
✓ Solution (S-II):
Data-recovery-based Session Complement

- Distinguish false splits vs. missing startup/shutdown patterns.
- Insert missing RouterInfos contextually.
- Reconstruct accurate online session boundaries.

Outcome: Accurate reconstruction of online sessions even with only partial RouterInfo observations.

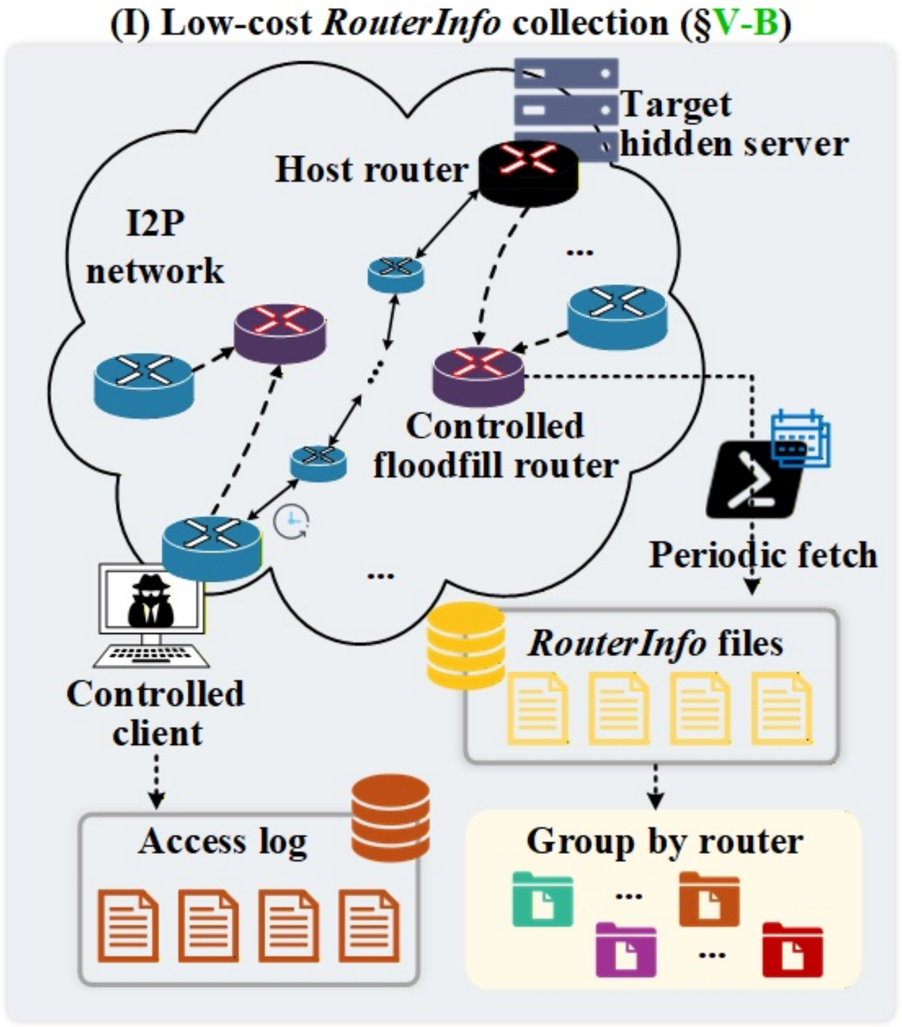
I2Perception Overview

Passive data collection + live behavior inference + behavior alignment



Passive Data Collection

Low-cost RouterInfo monitoring using floodfill routers



Deploy a few floodfill routers

↓

Other I2P routers periodically upload RouterInfo to floodfill routers

↓

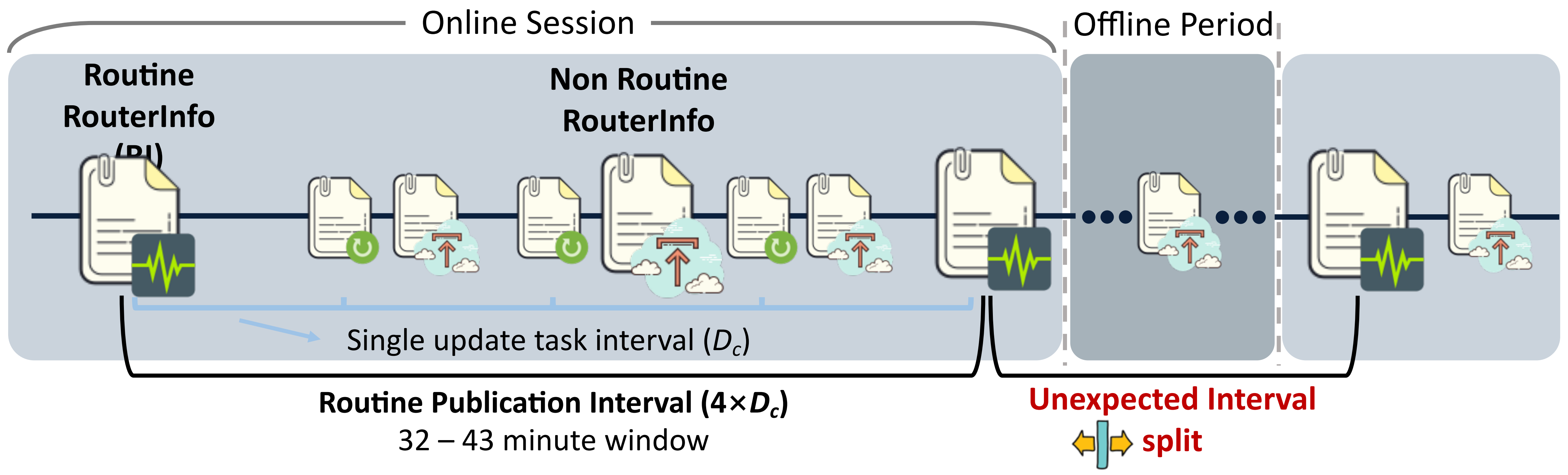
We passively receive and store these publications

↓

Construct RouterInfo traces for each router

Live Behavior Inference — Step 1: Coarse-grained session inference

Detect offline gaps from routine publication



1. Routine RI Extraction

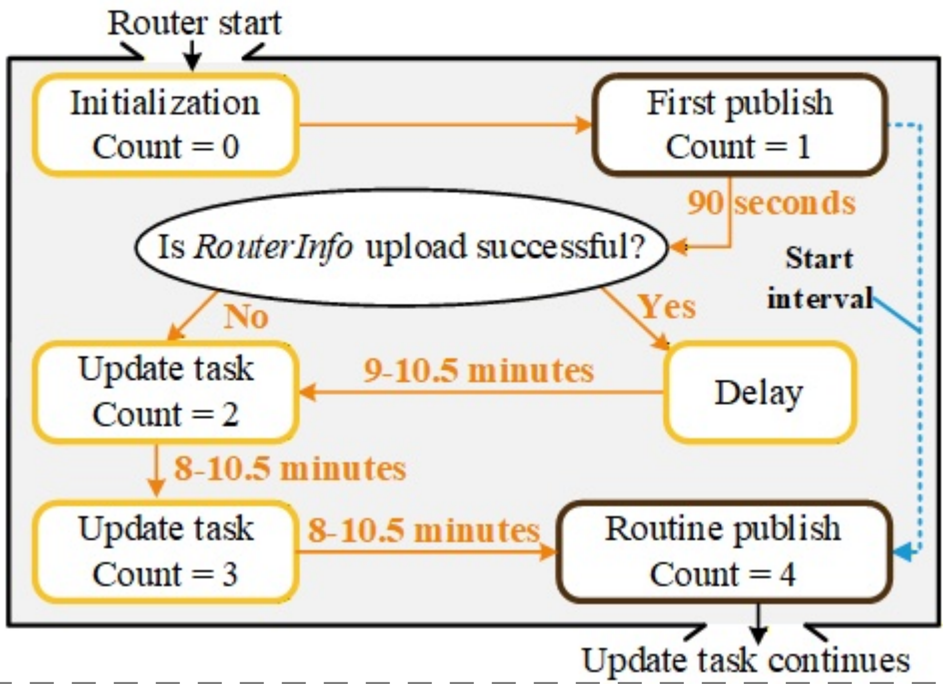
Find all routine RI from the raw data trace using **routine publication interval**.

2. Online Session Construction

Merge consecutive routine RI into a single **online session**; gaps between sessions are treated as **offline periods**.

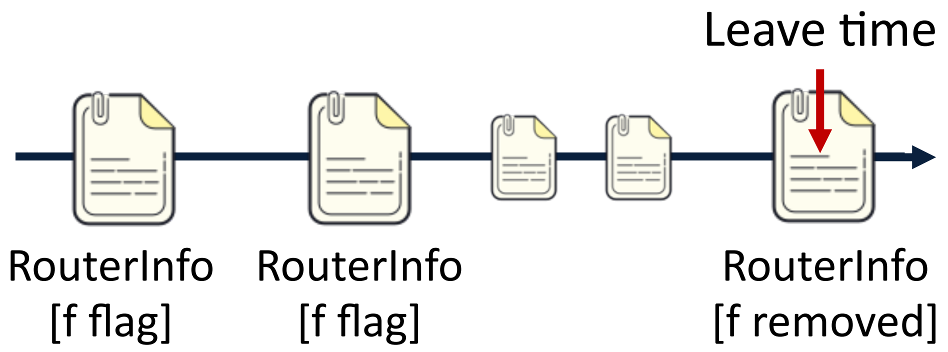
Live Behavior Inference — Step 2: Join and leave behavior identification

Identifying router join and leave times by matching specific RouterInfo publication signatures.



What is the join behavior?

- An time interval of 3 × D_c (25-31.5 minutes) or 90 + 2 × D_c (17.5-22.5 minutes) between the first RouterInfo and first routine RouterInfo.

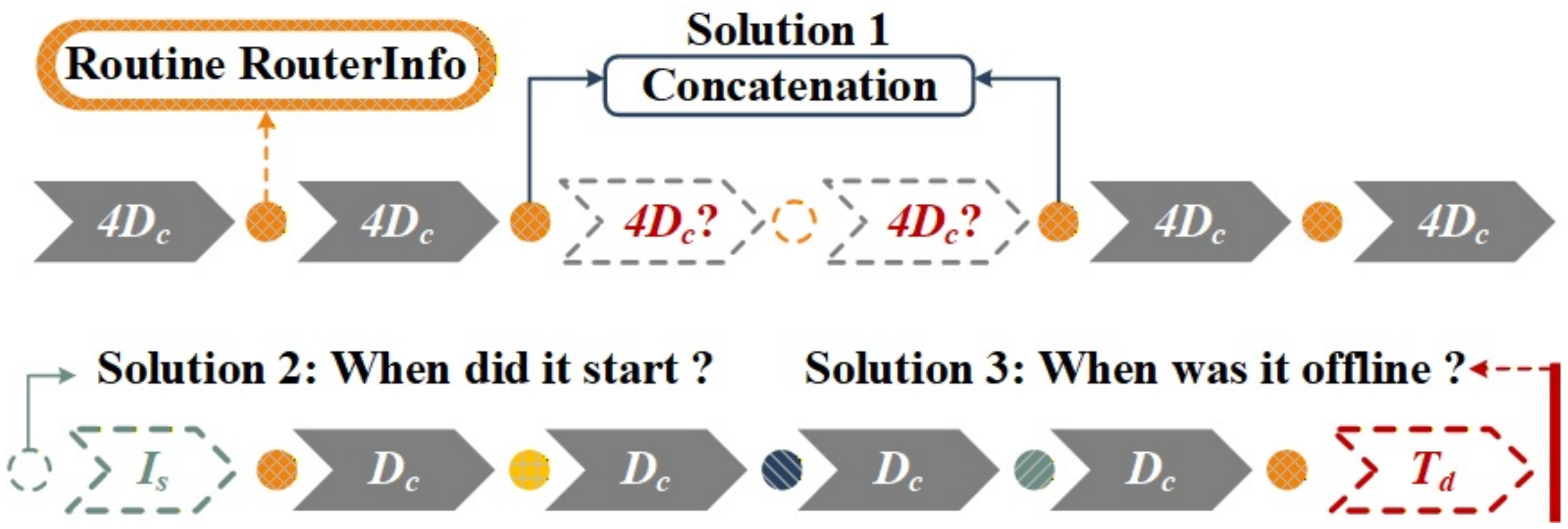


What is the leave behavior?

- The final RouterInfo published by a floodfill router before shutdown, where the 'f' flag is removed to indicate termination of floodfill service.

Live Behavior Inference — Step 3: Online session complement

Recover sessions under incomplete RouterInfo collections

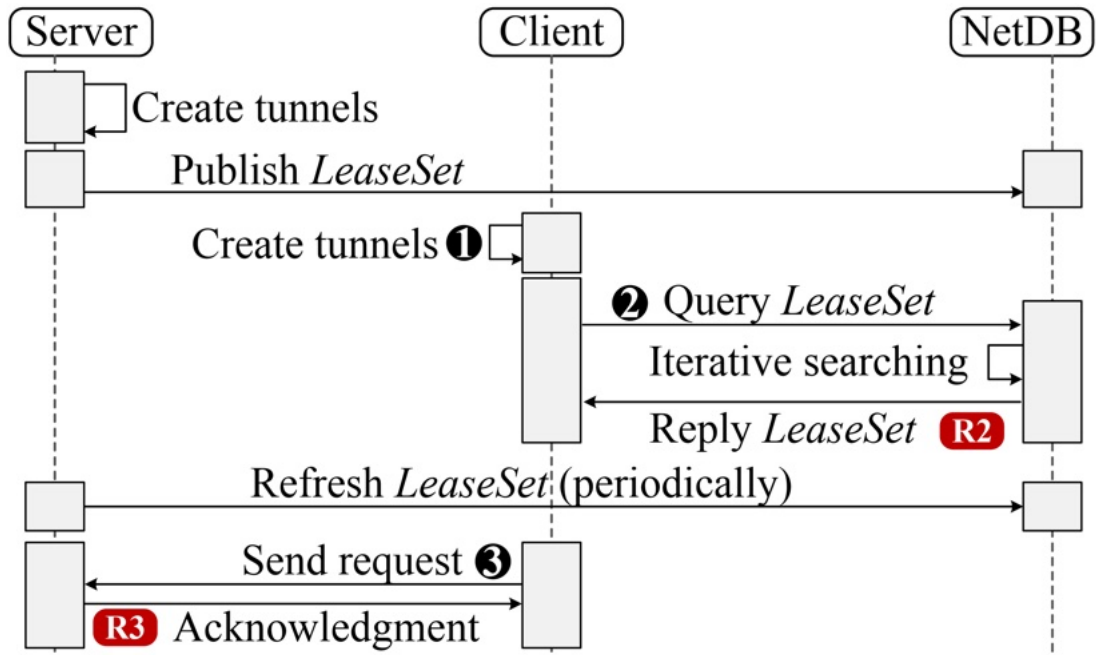


Three complement solutions

1) Based on routine RI publication	->	Session concatenation
2) Based on join behavior	->	Startup time recovery.
3) Based on leave behavior	->	Offline time estimation.

Active Probing: Infer Hidden-service Live Behavior

Obtain the service on–off trace from the client side



Three-stage probing process

1 Tunnel Establishment
Client builds inbound/outbound tunnels.

2 LeaseSet Lookup (R2)
- R2 failure → LeaseSet expired → **service offline > 10 min**
- R2 success → LeaseSet present

3 Service Connection (R3)
- R3 received → **service online**
- No R3 when LeaseSet present → **offline < 10 min**

- Deploy an I2P client that periodically accesses the target service
- Record success/failure timestamps (availability over time)
- Convert to an on–off behavior sequence aligned to router session granularity

Evaluation — Setup

Real-world monitoring all I2P routers and controlled hidden services

Deployment

- 15 controlled floodfill routers for RouterInfo collection
- 10 controlled host routers running hidden services (5 Java, 5 C++)
- Monitor for over 8 months to validate long-term effectiveness
- One malicious I2P client probes hidden services and runs deanonymization prototype

Hidden service behavior scenarios

S1–S4: frequent on–off (minute level)

- S1: 45–60 min offline gaps
- S2: 30-45 min offline gaps
- S3: 5–10 min offline gaps
- S4: 5-60 min mix offline gaps

S5–S7: long-lived (day level)

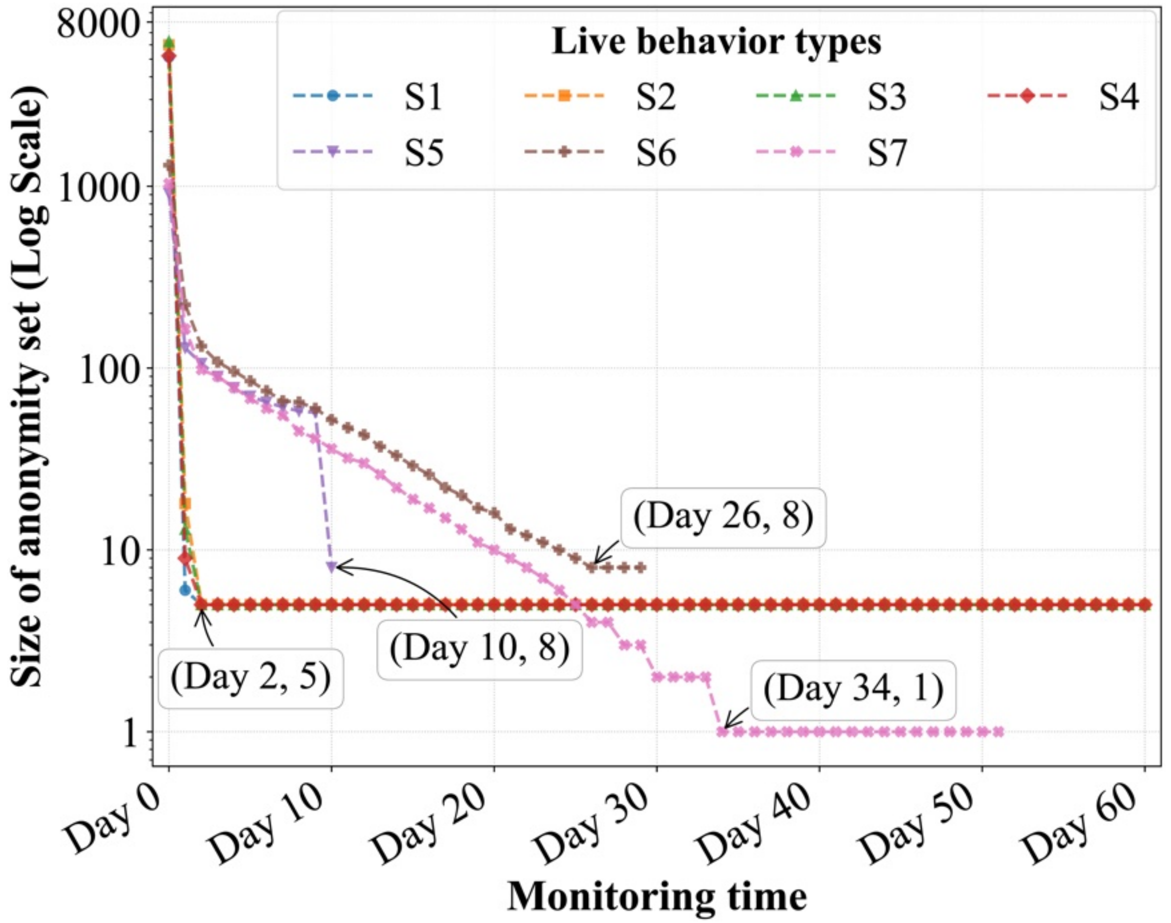
- S5: online 9 days -> offline on day 10
- S6: online 28 days -> offline on day 29
- S7: online 50 days → offline on day 51

Metrics

- Join/leave inference bias (seconds)
- Anonymity set size ***k*** over time
- Time to deanonymize

Results: Deanonymizing Controlled Hidden Services

Anonymity sets shrink rapidly with monitoring



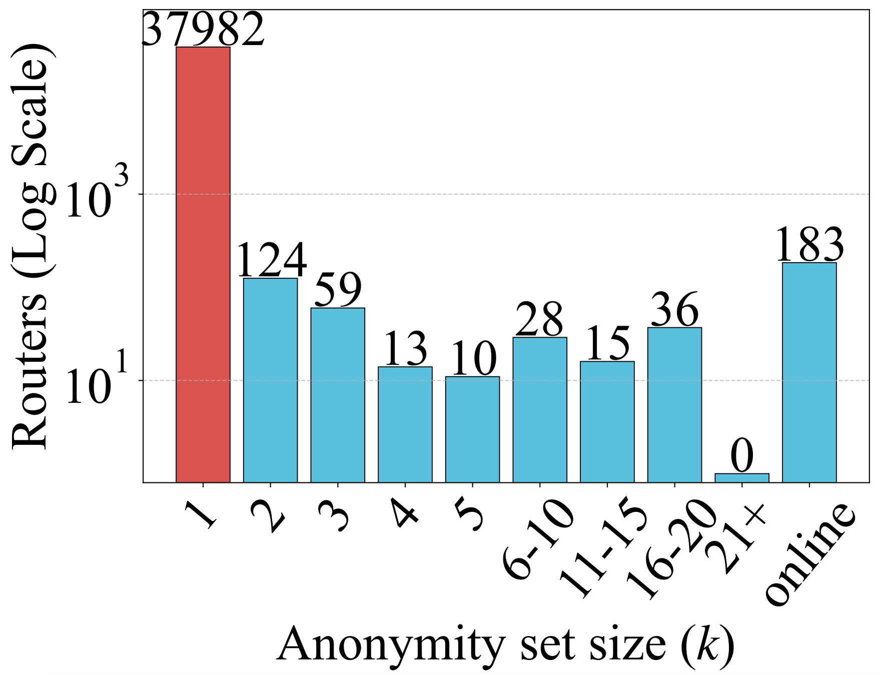
Key findings (controlled services)

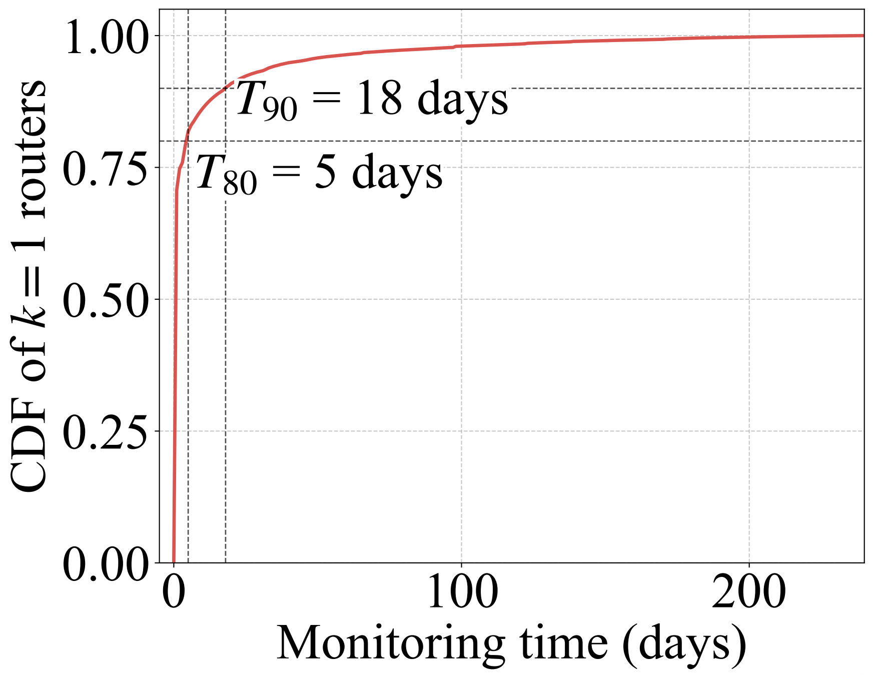
- Short-lived scenarios (S1–S4): k shrinks from around 5,000 - 6,000 to around 10 on Day 1, and from Day 2 onward, k reaches 5 (the number of synchronized services in the experiment)
- Long-lived scenarios (S5 and S6): *k* drops from 900 - 1,300 to around 129 - 223 after one day, then continues shrinking
- Single service in S7 is deanonymized after 34 days of monitoring

Summarize: I2PERCEPTION successfully de-anonymizes all controlled hidden services.

Results: Behavior Uniqueness Appears Quickly

Most routers become distinguishable over time





Results

- Among routers monitored from day 1, **37,982** eventually exhibit unique behaviors.
- Only **468** share similar behavior with at least one other router within the **8 months**.
- **90%** of unique routers become distinguishable within the first **18 days**.
- For non-unique routers, anonymity set still shrinks **from thousands to tens**.

Mitigations

Reduce RouterInfo leakage or disrupt behavior alignment

Protocol-Level Defenses

- **Randomize** RouterInfo publication (10–55 minute window).
- Remove deterministic publication patterns during join and leave phase.

User-Side Measures

- Synchronize behaviors across multiple hidden services.
- Detect repeated / periodic probing via I2P identities and throttle or ignore suspicious clients.

Official patch **has been deployed** by I2P

Key Principle: Eliminate deterministic temporal patterns to prevent time-based deanonymization.

Conclusion

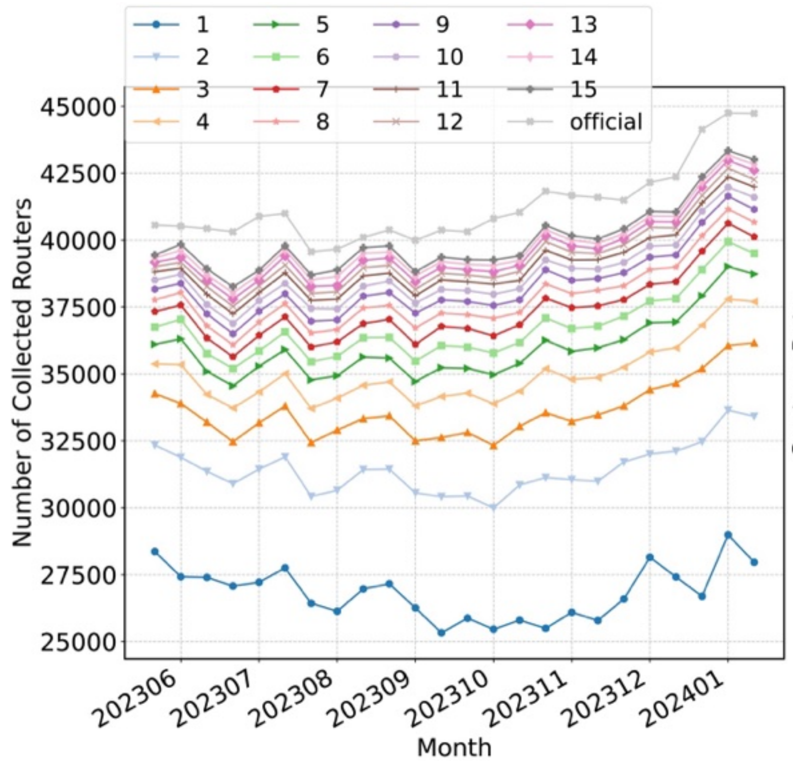
- User live behavior (on-off pattern) is a powerful side channel in I2P
- I2Perception: low-cost data collection + live behavior inference + behavior alignment
- Monitoring shrinks anonymity sets dramatically; most routers have unique behaviors
- Mitigations: randomize RouterInfo publication + remove join/leave behaviors

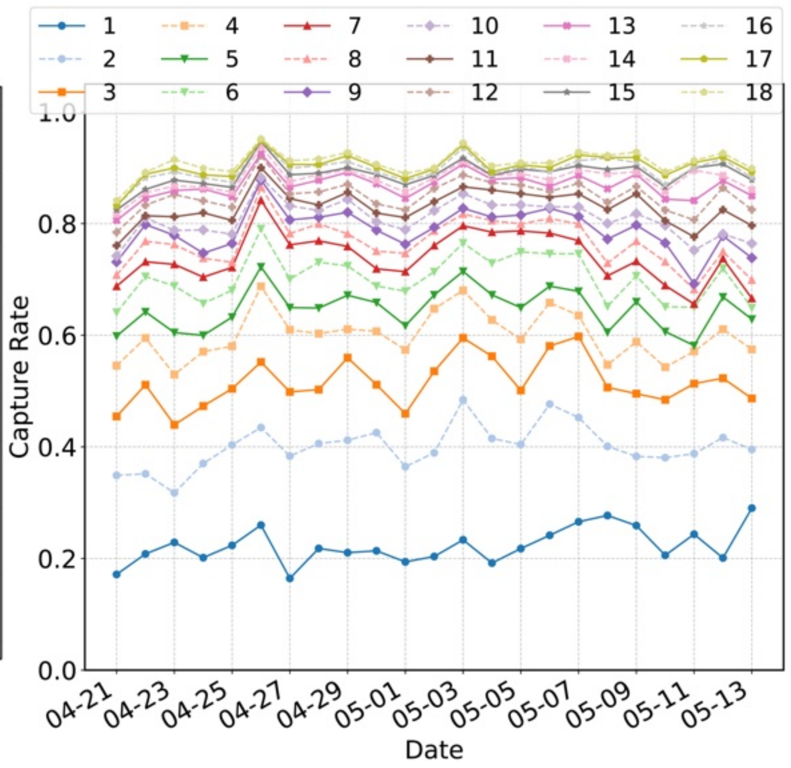
Thanks **Q&A**

Contact: <wanghongze@seu.edu.cn> • Extend version: arXiv:2512.15510

Passive Data Collection

The performance of deploying 15 floodfill routers.





Network-wide coverage

98%

vs officially reported router count

Average RouterInfo capture rate

90%

comparable to larger deployments

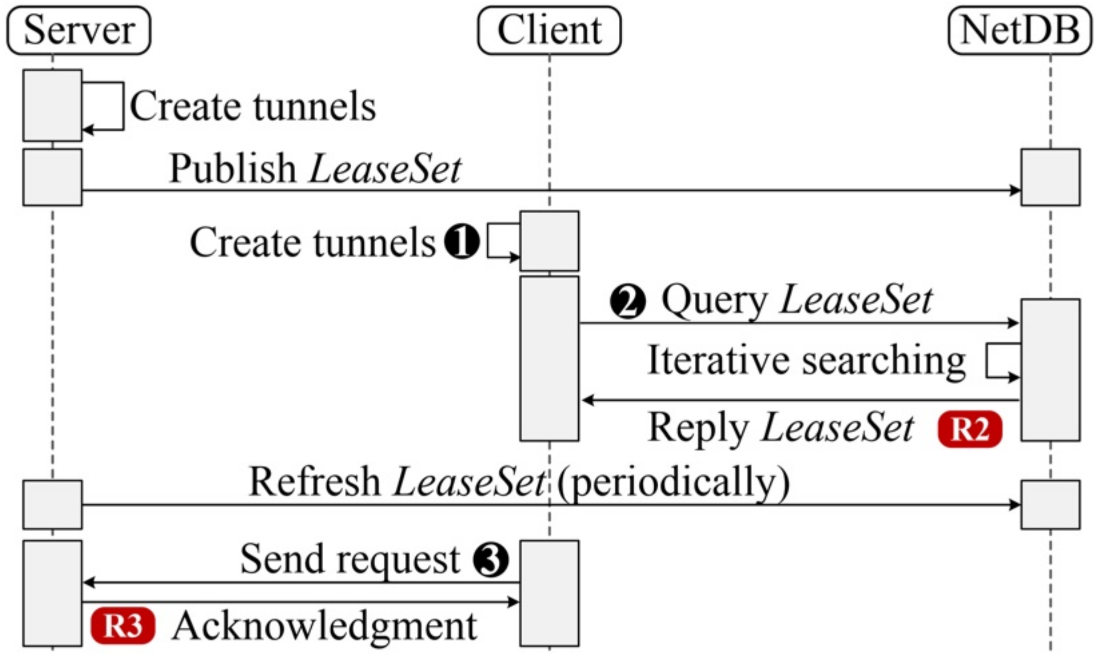
Monitoring window used in study

8+ months

persistent observation

Active Probing: Infer Hidden-service Live Behavior

Obtain the service on–off trace from the client side



Three-stage probing process

1 Tunnel Establishment
Client builds inbound/outbound tunnels.

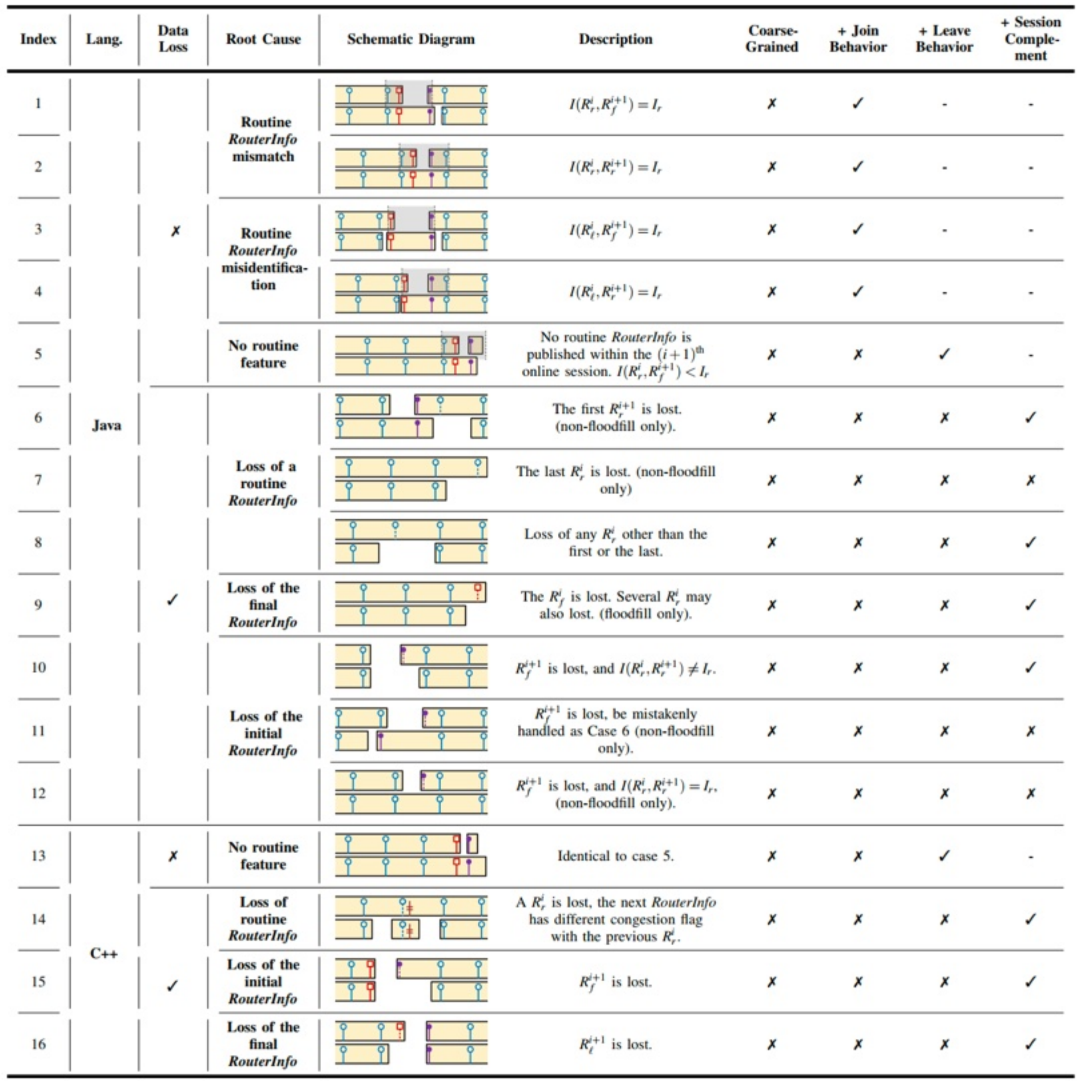
2 LeaseSet Lookup (R2)
- R2 failure → LeaseSet expired → **service offline > 10 min**
- R2 success → LeaseSet present

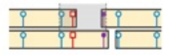
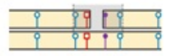
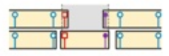
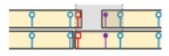
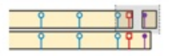
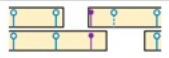
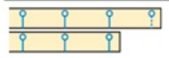
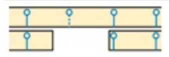
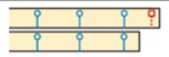
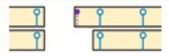
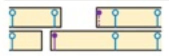
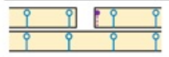
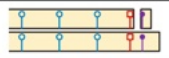
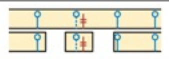
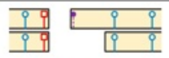
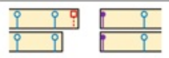
3 Service Connection (R3)
- R3 received → **service online**
- No R3 when LeaseSet present → **offline < 10 min**

Why probing is reliable and stealthy
- I2P ensures reliable delivery (SSU / NTCP + streaming layer).
- As long as RouterInfo and service responses are deliverable, probing outcomes accurately reflect service liveness.
- Client anonymity is preserved by I2P.

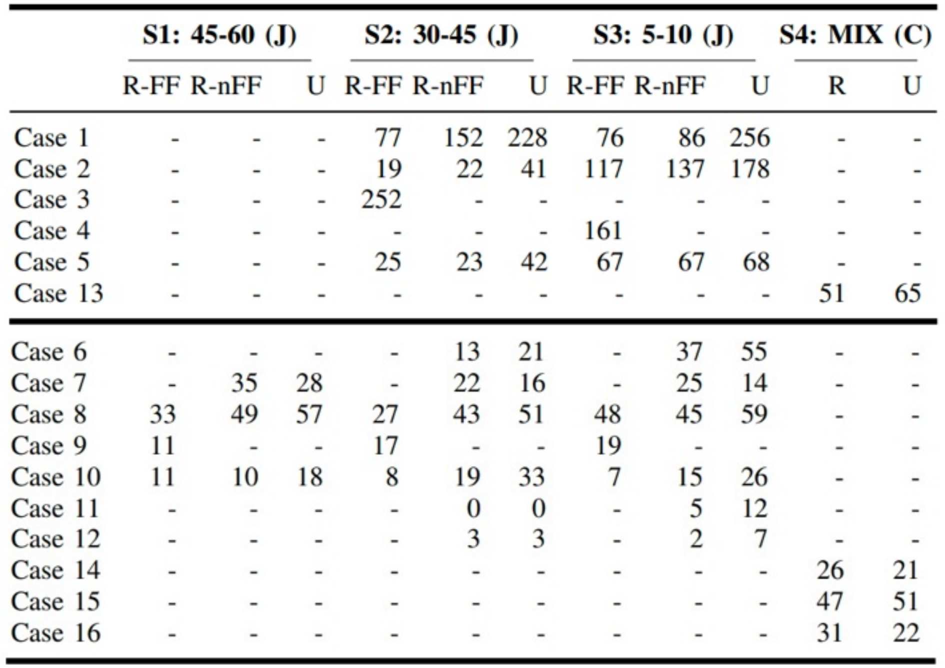
Results: Why All Live Behavior Inference Steps Are Necessary

Comprehensive Enumeration and Empirical Occurrence of Challenging Cases



Index	Lang.	Data Loss	Root Cause	Schematic Diagram	Description	Coarse-Grained	+ Join Behavior	+ Leave Behavior	+ Session Complement
1			**Routine *RouterInfo* mismatch**		$I(R_r^i, R_f^{i+1}) = I_r$	✗	✓	-	-
2					$I(R_r^i, R_r^{i+1}) = I_r$	✗	✓	-	-
3		✗	**Routine *RouterInfo* misidentification**		$I(R_\ell^i, R_f^{i+1}) = I_r$	✗	✓	-	-
4					$I(R_\ell^i, R_r^{i+1}) = I_r$	✗	✓	-	-
5			**No routine feature**		No routine *RouterInfo* is published within the $(i+1)^{\text{th}}$ online session. $I(R_r^i, R_f^{i+1}) < I_r$	✗	✗	✓	-
6	**Java**				The first R_r^{i+1} is lost. (non-floodfill only).	✗	✗	✗	✓
7			**Loss of a routine *RouterInfo***		The last R_r^i is lost. (non-floodfill only)	✗	✗	✗	✗
8					Loss of any R_r^i other than the first or the last.	✗	✗	✗	✓
9		✓	**Loss of the final *RouterInfo***		The R_f^i is lost. Several R_r^i may also lost. (floodfill only).	✗	✗	✗	✓
10					R_f^{i+1} is lost, and $I(R_r^i, R_r^{i+1}) \neq I_r$.	✗	✗	✗	✓
11			**Loss of the initial *RouterInfo***		R_f^{i+1} is lost, be mistakenly handled as Case 6 (non-floodfill only).	✗	✗	✗	✗
12					R_f^{i+1} is lost, and $I(R_r^i, R_r^{i+1}) = I_r$, (non-floodfill only).	✗	✗	✗	✗
13		✗	**No routine feature**		Identical to case 5.	✗	✗	✓	-
14	**C++**		**Loss of routine *RouterInfo***		A R_r^i is lost, the next *RouterInfo* has different congestion flag with the previous R_r^i.	✗	✗	✗	✓
15		✓	**Loss of the initial *RouterInfo***		R_f^{i+1} is lost.	✗	✗	✗	✓
16			**Loss of the final *RouterInfo***		R_ℓ^{i+1} is lost.	✗	✗	✗	✓

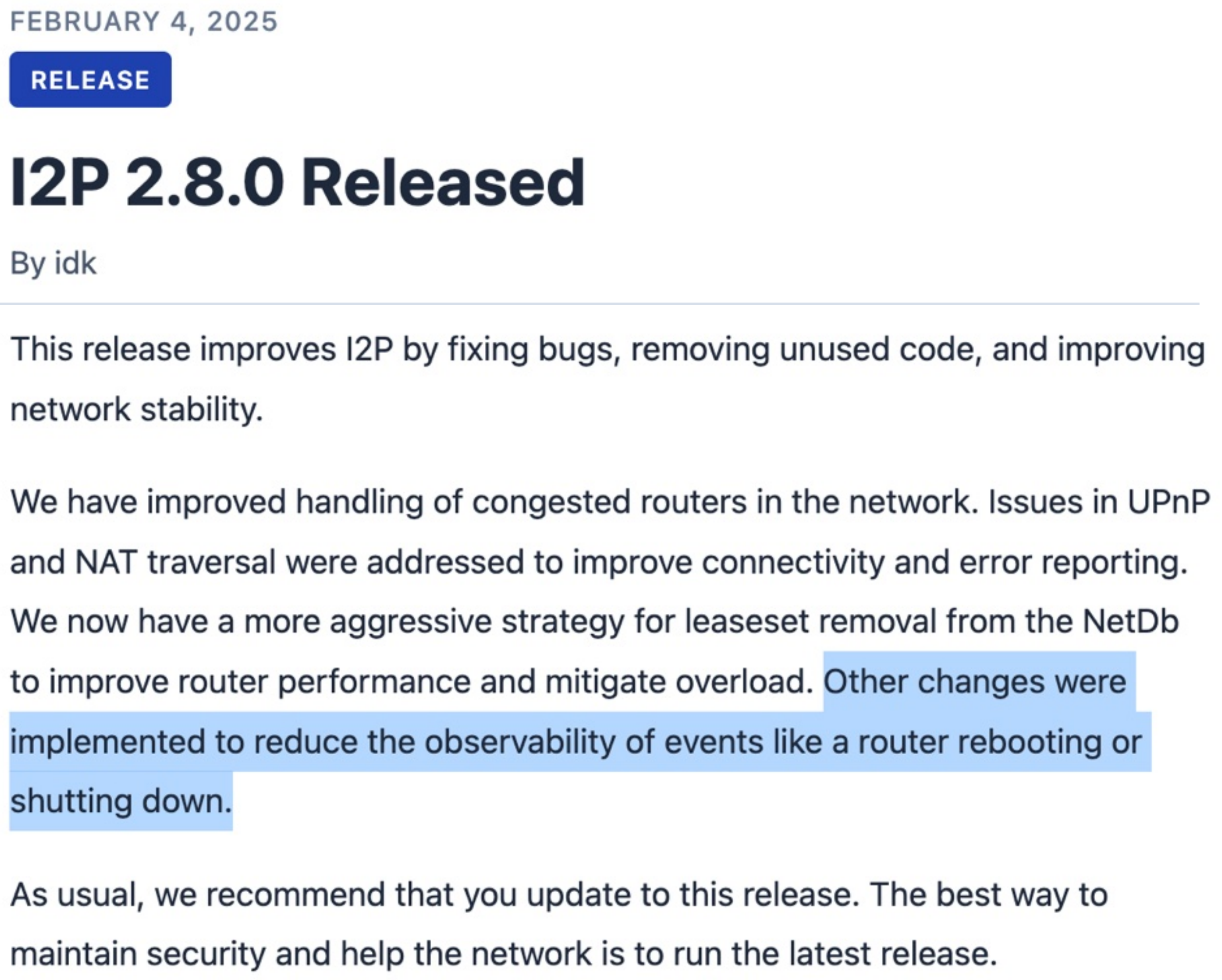
- We systematically enumerate all RouterInfo publication cases that may **challenge live behavior inference**.
- Controlled experiments were conducted to trigger each case. Real-world measurements confirm that **these cases naturally occur**.



	S1: 45-60 (J)			S2: 30-45 (J)			S3: 5-10 (J)			S4: MIX (C)	
	R-FF	R-nFF	U	R-FF	R-nFF	U	R-FF	R-nFF	U	R	U
Case 1	-	-	-	77	152	228	76	86	256	-	-
Case 2	-	-	-	19	22	41	117	137	178	-	-
Case 3	-	-	-	252	-	-	-	-	-	-	-
Case 4	-	-	-	-	-	-	161	-	-	-	-
Case 5	-	-	-	25	23	42	67	67	68	-	-
Case 13	-	-	-	-	-	-	-	-	-	51	65
Case 6	-	-	-	-	13	21	-	37	55	-	-
Case 7	-	35	28	-	22	16	-	25	14	-	-
Case 8	33	49	57	27	43	51	48	45	59	-	-
Case 9	11	-	-	17	-	-	19	-	-	-	-
Case 10	11	10	18	8	19	33	7	15	26	-	-
Case 11	-	-	-	-	0	0	-	5	12	-	-
Case 12	-	-	-	-	3	3	-	2	7	-	-
Case 14	-	-	-	-	-	-	-	-	-	26	21
Case 15	-	-	-	-	-	-	-	-	-	47	51
Case 16	-	-	-	-	-	-	-	-	-	31	22

Responsible Disclosure & Ethics

Ethical safeguards and disclosure process



FEBRUARY 4, 2025

RELEASE

I2P 2.8.0 Released

By idk

This release improves I2P by fixing bugs, removing unused code, and improving network stability.

We have improved handling of congested routers in the network. Issues in UPnP and NAT traversal were addressed to improve connectivity and error reporting. We now have a more aggressive strategy for leaseset removal from the NetDb to improve router performance and mitigate overload. Other changes were implemented to reduce the observability of events like a router rebooting or shutting down.

As usual, we recommend that you update to this release. The best way to maintain security and help the network is to run the latest release.

Ethical Considerations

- **Controlled Deployment:** All target hidden services were deployed and operated by us; no real user services were involved.
- **Compliance with I2P Research Guidelines:** No active exploits, DoS attacks, social engineering, or attacks against I2P infrastructure.
- **Data Protection & Minimal Impact**
  - Data collected only via 15 self-deployed floodfill routers.
  - Passive observation; no modification of network data.
  - Offline analysis; no third-party sharing.
  - Encrypted transmission, secure storage, anonymized and deleted after submission.