FRTR: A Scalable Mechanism for Global Routing Consistency

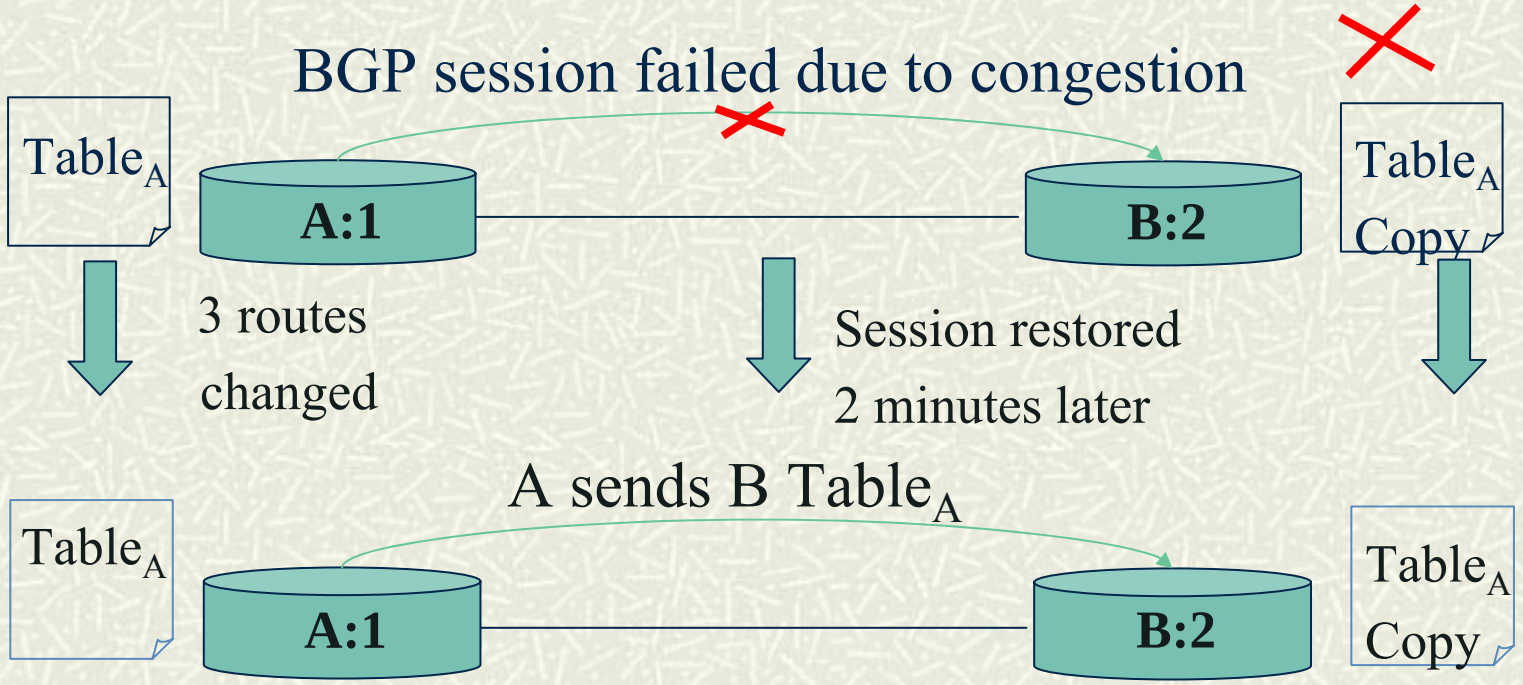
Lan Wang[Ŧ], Nischal Piratla[*], Dan Massey[*], Keyur Patel[t] and Lixia Zhang[§]

[Ŧ]Univ. of Memphis, [*]Colorado State Univ.,

[t]Cisco Systems, [§]UCLA

Why does BGP need fast routing recovery?

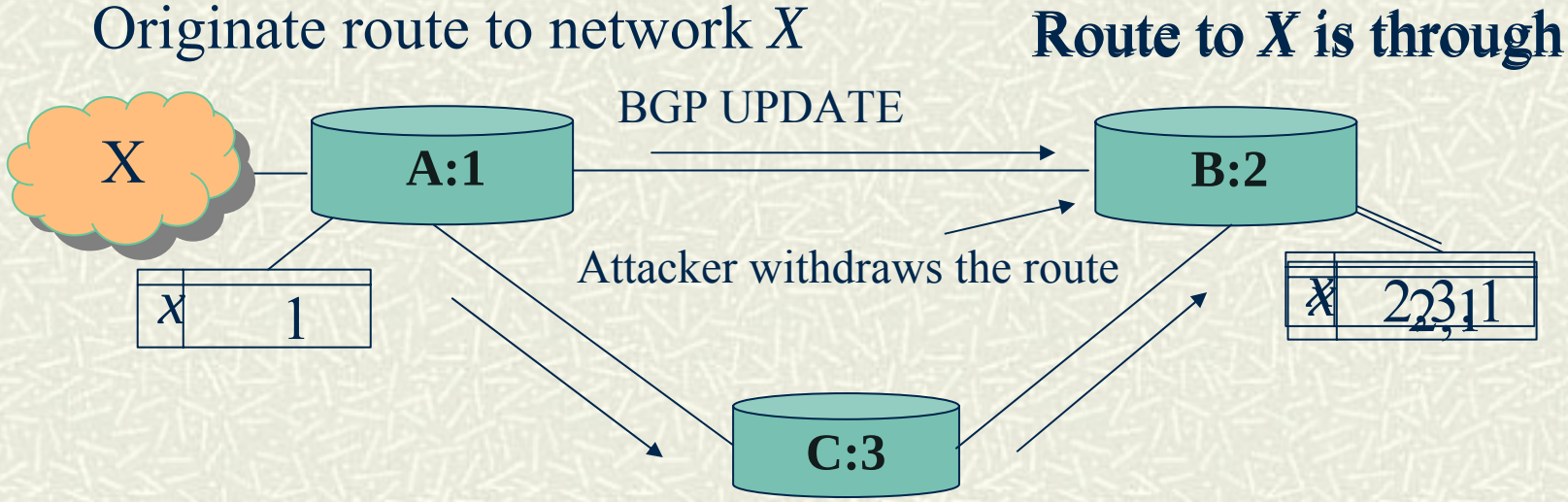
BGP recovers inefficiently from transient session resets.



3 routes were changed, but A sent 120K+ routes (~5MB).

Why does BGP need fast routing recovery? (cont'd)

Faults and attacks can corrupt a routing table.



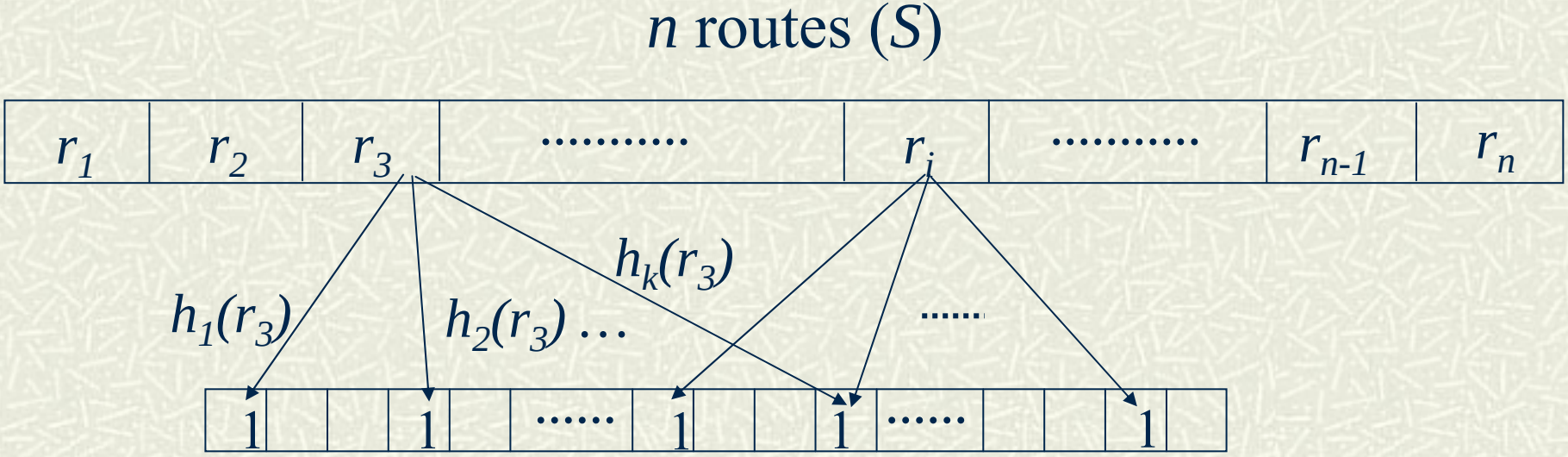
Attacker has successfully changed B's route to X!

Other fault sources:
hardware failures, human errors, software bugs

FRTR (Fast Routing Table Recovery)

- Use small Bloom filter to encode routing table;
- Exchange digests periodically.

Bloom Filter Digest



l/*n*: encoding ratio, a small constant, e.g. 5

$h_1()$, $h_2()$, …, $h_k()$: hash functions

Is *r* a member of *S*? $\Rightarrow$ check d($h_j(r)$) = 1? (j = 1, 2, …, k)

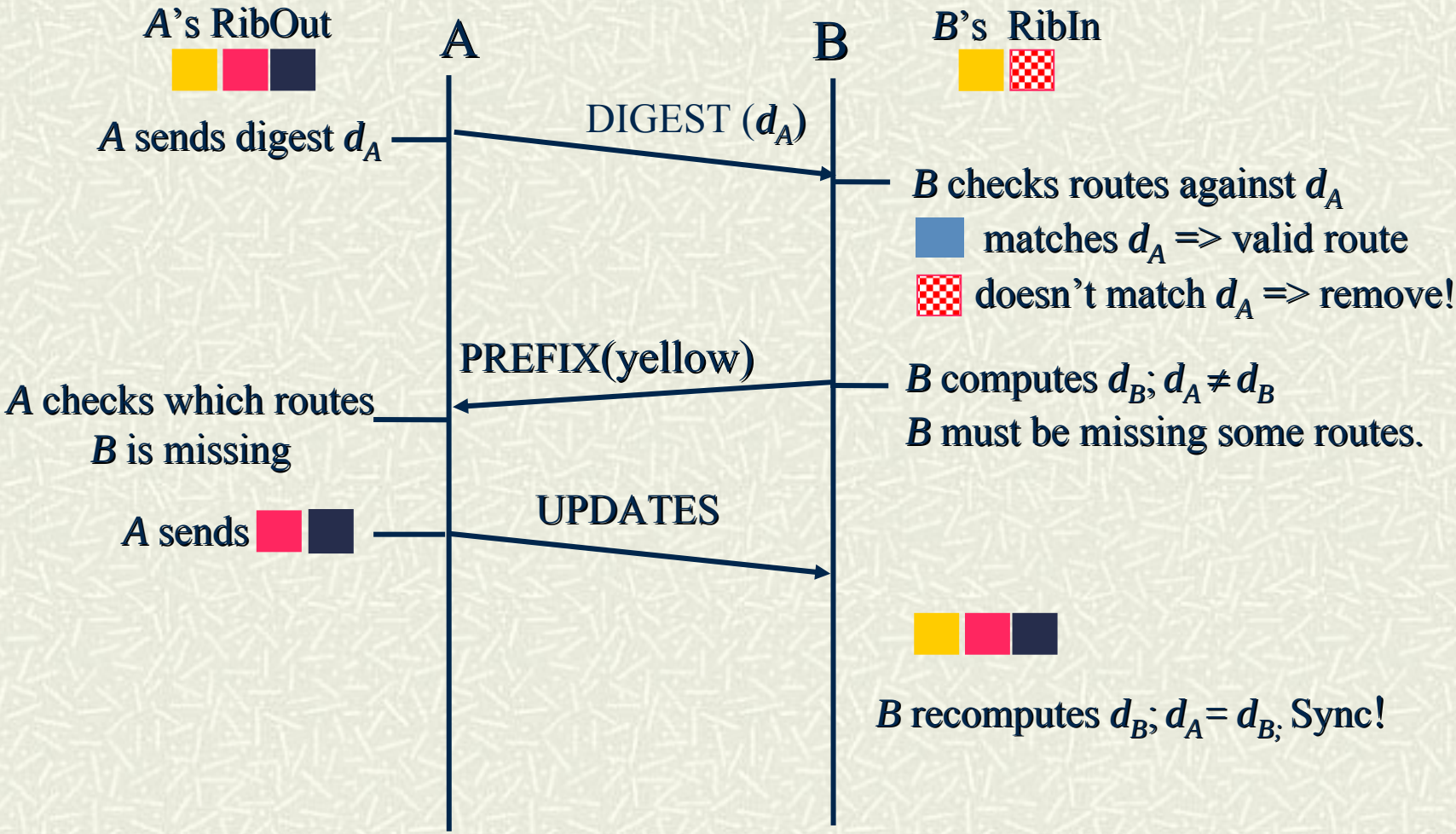
Why use Bloom Filter?

- Compact size $\Rightarrow$ low bandwidth cost
- Low false positive rate
  - a *small* probability f that an invalid route will be considered valid.
  - $f = (1 - e^{-k \cdot n/l})^k$ (e.g. $l/n = 8, k = 3 \Rightarrow f = 3.1\%$)

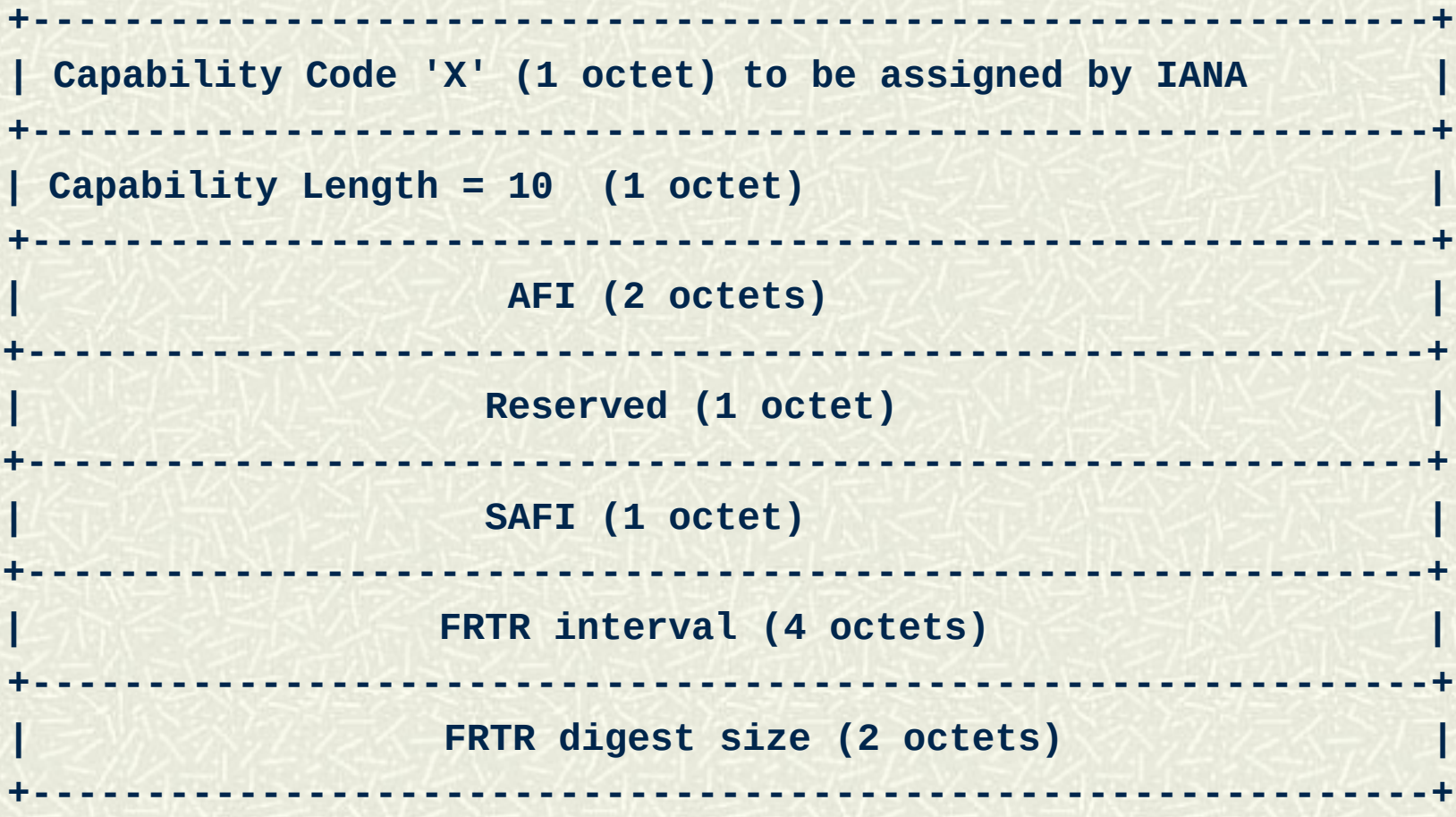
How to remove the false positives?

- Good news: some false positives may be exposed *as the state set evolves*.
- For the remaining false positives, take advantage of the periodic digests.
  - Use seeded hash functions, e.g. salted MD5;
  - Change the seed periodically;
- Residual error $e = f^i$ (i = # different seeds)
  - Example: 99.997% of the inconsistencies will be removed when $f = 3.1\%$ and $i = 3$.

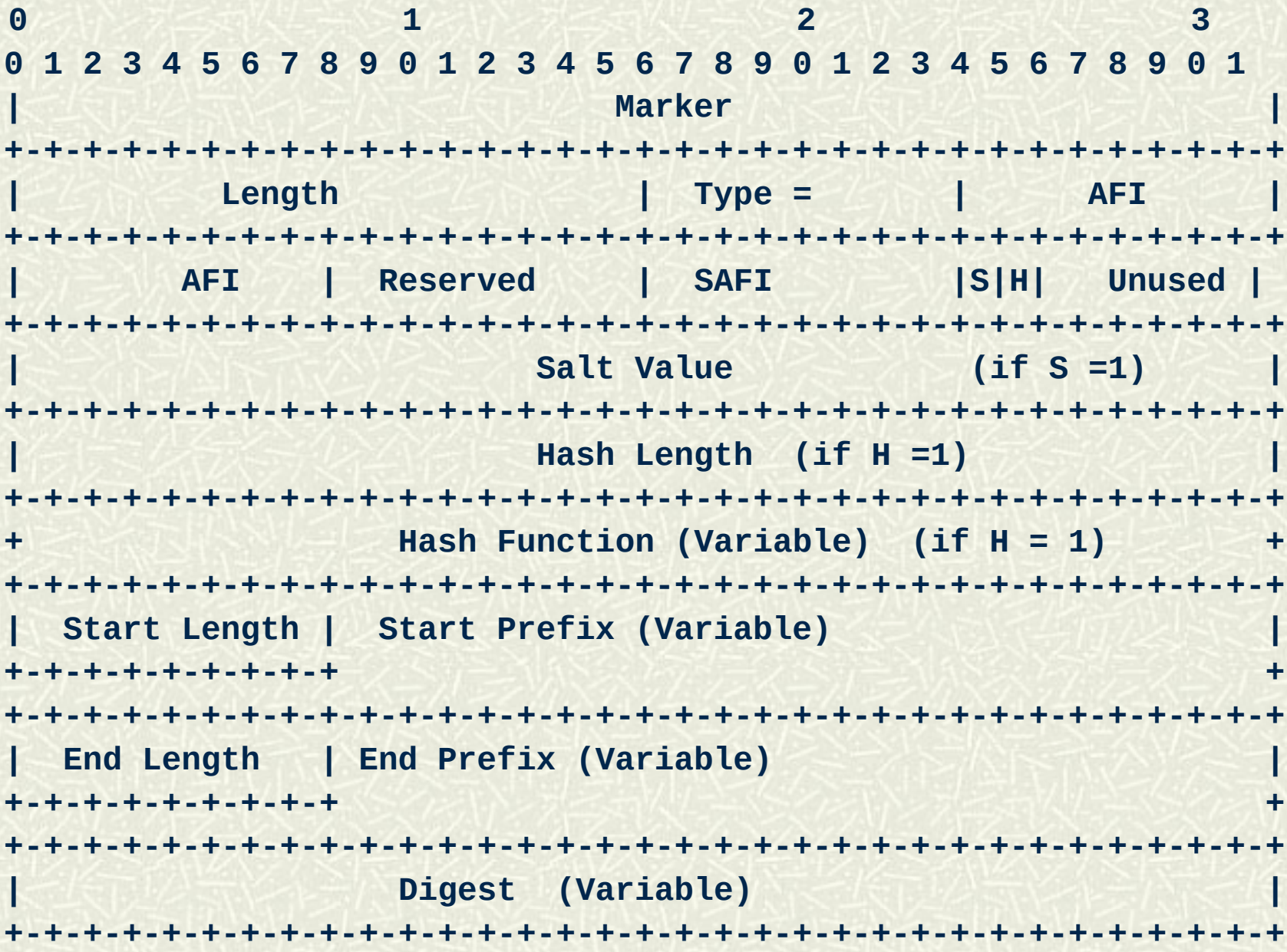
Routing Table Recovery Procedure



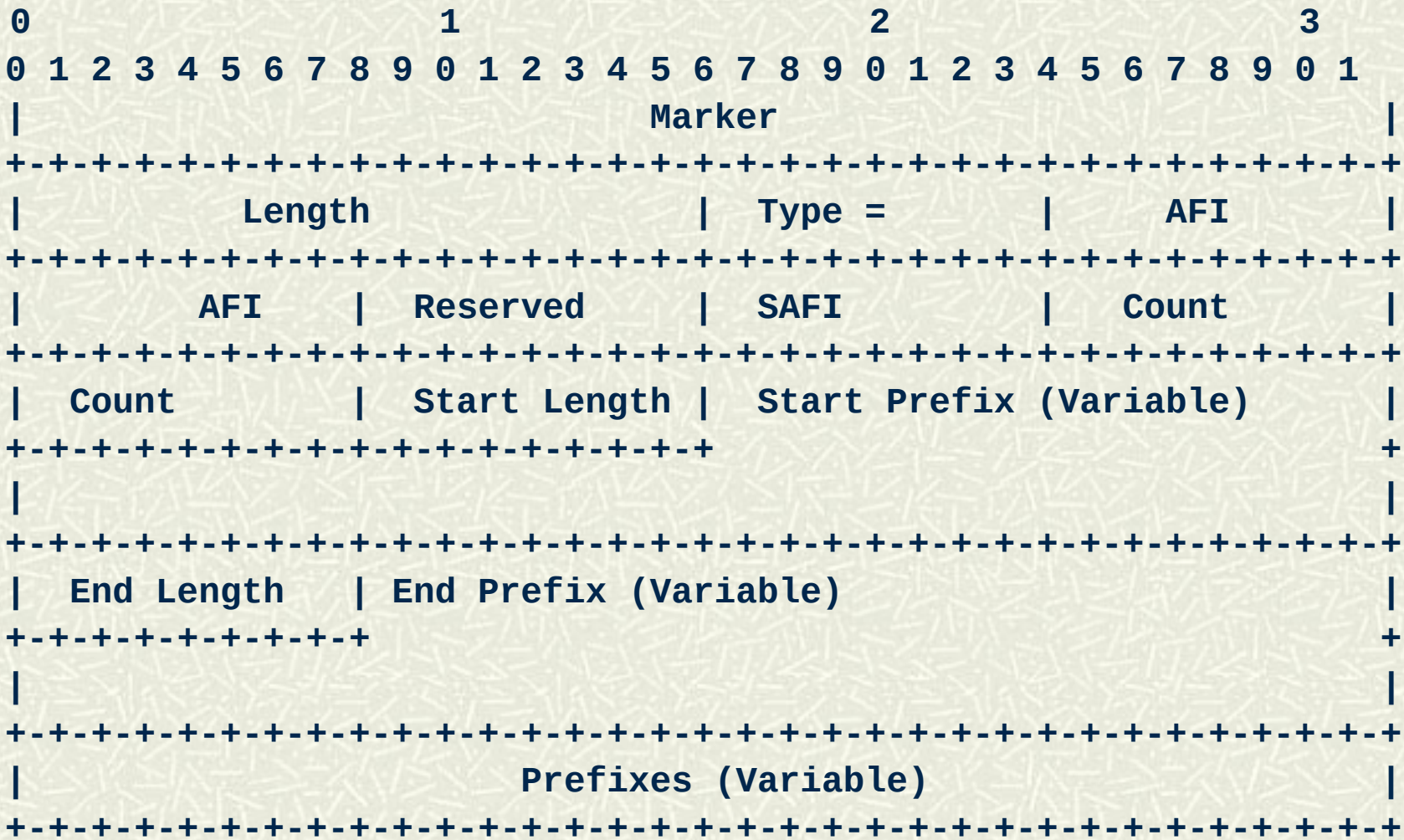
BGP OPEN Message – FRTR Negotiation



Digest Message



Prefix Message

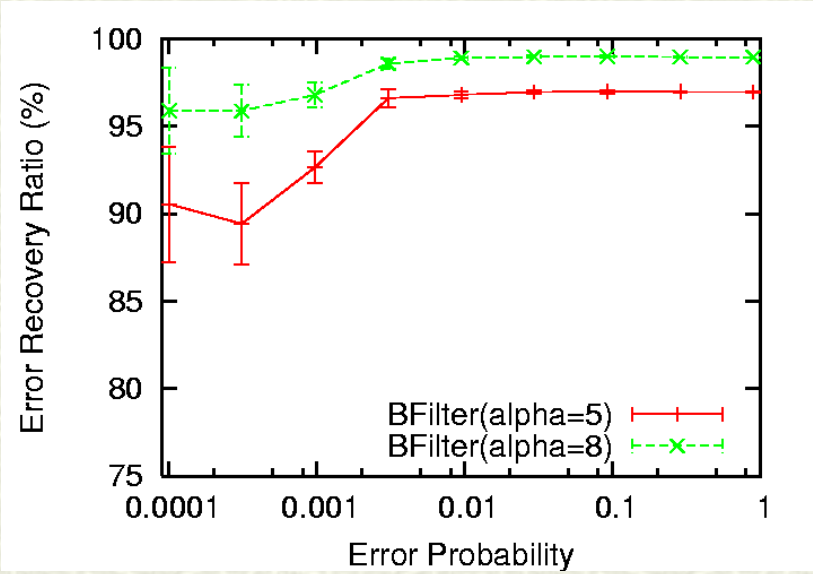


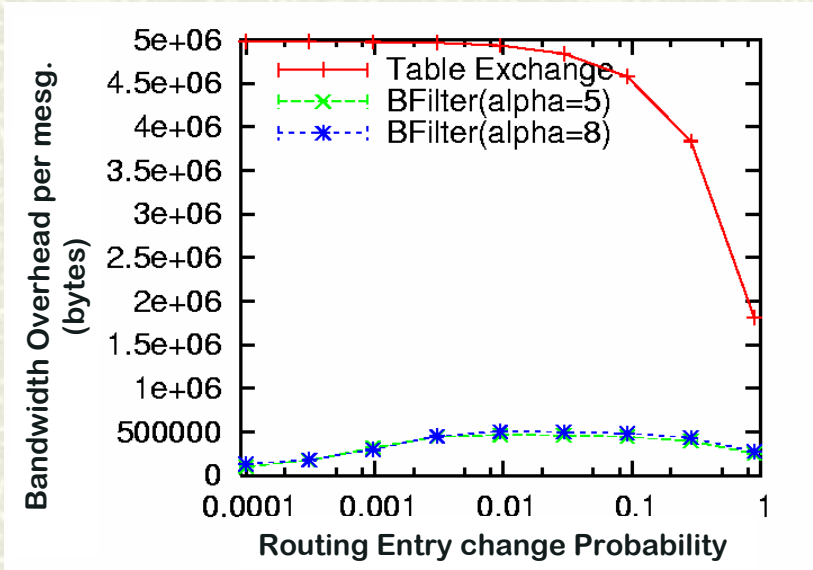
Evaluation of FRTR

- Data source: routing tables from 11 ISPs
- Methodology
  - Inject random errors into each routing table;
  - Simulate two BGP peers: one with a original table and the other with a corrupted table.
- Bloom Filter Parameters
  - 3 hash functions
  - Encoding ratio: 5 and 8

FRTR Performance

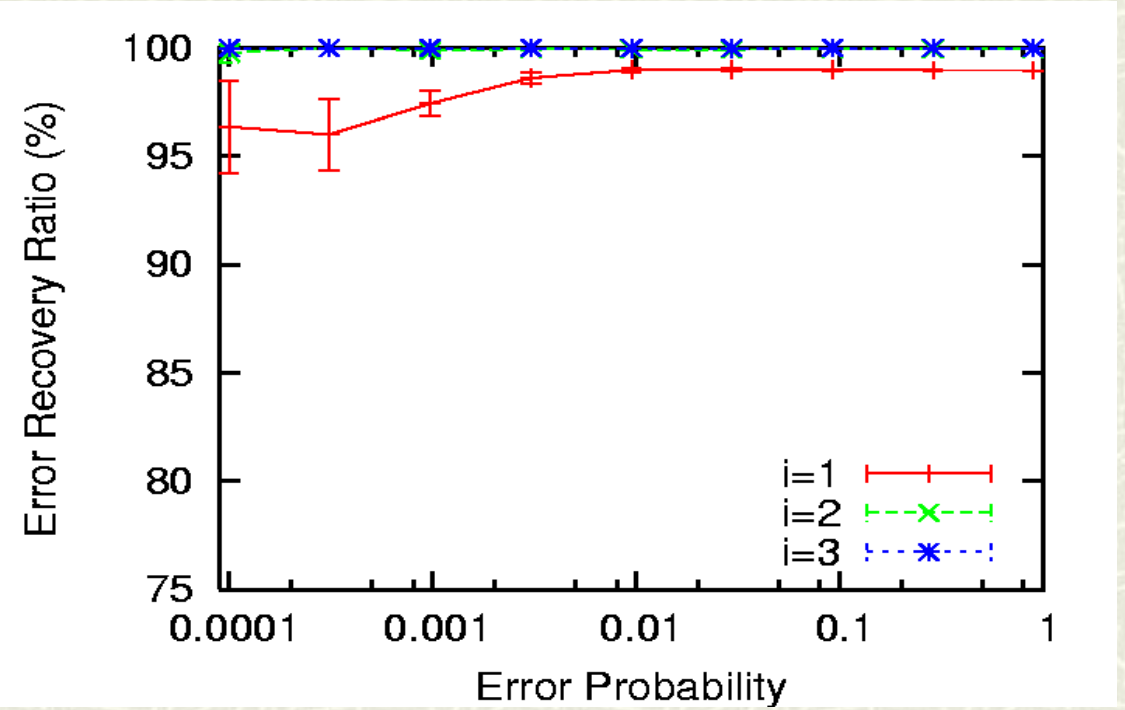
- Error recovery ratio: 91%~97% for encoding ratio of 5, 96% ~ 99% for encoding ratio of 8
- Bandwidth overhead per message (encoding ratio = 5): as low as 75K bytes, at most 460K bytes





Performance of Salted Digests

- Error recovery ratio (encoding ratio = 8): close to 100% after a few seed changes.



Summary

- FRTR encodes routing table to efficiently detect and recover unnoticeable errors.
- FRTR significantly reduces the overhead of routing table recovery after session resets.