NDSS SYMPOSIUM/2024

Symphony: Path Validation at Scale

Anxiao He, Jiandong Fu, Kai Bu, Ruiqi Zhou, Chenlu Miao, Kui Ren

浙江大学 ZHEJIANG UNIVERSITY

Background: Packet Forwarding Security

- Potential attack surfaces

Traffic diversion

Attacker eavesdrops any parts of packets with potentially sensitive information

Fictitious premium path usage

ISPs use inferior path but charge for premium path

Packet injection with spoofed source address

Routers inject extra packets to incriminate source

Background: Packet Forwarding Security

- Potential attack surfaces
- Root causes

End hosts have no control over the paths that their packets take

End hosts have no verification of the actual path a packet took toward recipient

Background: Path Validation

- Path validation

Control

Select an intended path for a specific packet

Verification

Check if a packet traversed routers on its intended path in the correct order

Background: Path Validation

- Representative solutions

ICING (CoNEXT 2011)

Each router computes proofs for all its downstream routers and verifies the proofs of all its upstream routers

Background: Path Validation

- Representative solutions

ICING (CoNEXT 2011)

Each router computes proofs for all its downstream routers and verifies the proofs of all its upstream routers

OPT (SIGCOMM 2014)

Assume a trusted source that pre-computes all the proofs for each node; routers only need to verify and update corresponding proofs

Background: Path Validation

- Representative solutions

ICING (CoNEXT 2011)

Each router computes proofs for all its downstream routers and verifies the proofs of all its upstream routers

OPT (SIGCOMM 2014)

Assume a trusted source that pre-computes all the proofs for each node; routers only need to verify and update corresponding proofs

EPIC (USENIX Security 2020)

Following OPT, simplify computation and shorten proof size

Dilemma between Efficiency and Security

- Design dilemma

Proofs should be easy to compute for high efficiency

Proofs should be sufficiently secure to withstand attacks

- Efficiency barrier

There exists an inevitable lower bound of overhead for performing secure path validation

Dilemma between Efficiency and Security

- Design dilemma

Proofs should be easy to compute for high efficiency

Proofs should be sufficiently secure to withstand attacks

- Efficiency barrier

There exists an inevitable lower bound of overhead for performing secure path validation

- Root cause

Packet-wise validation

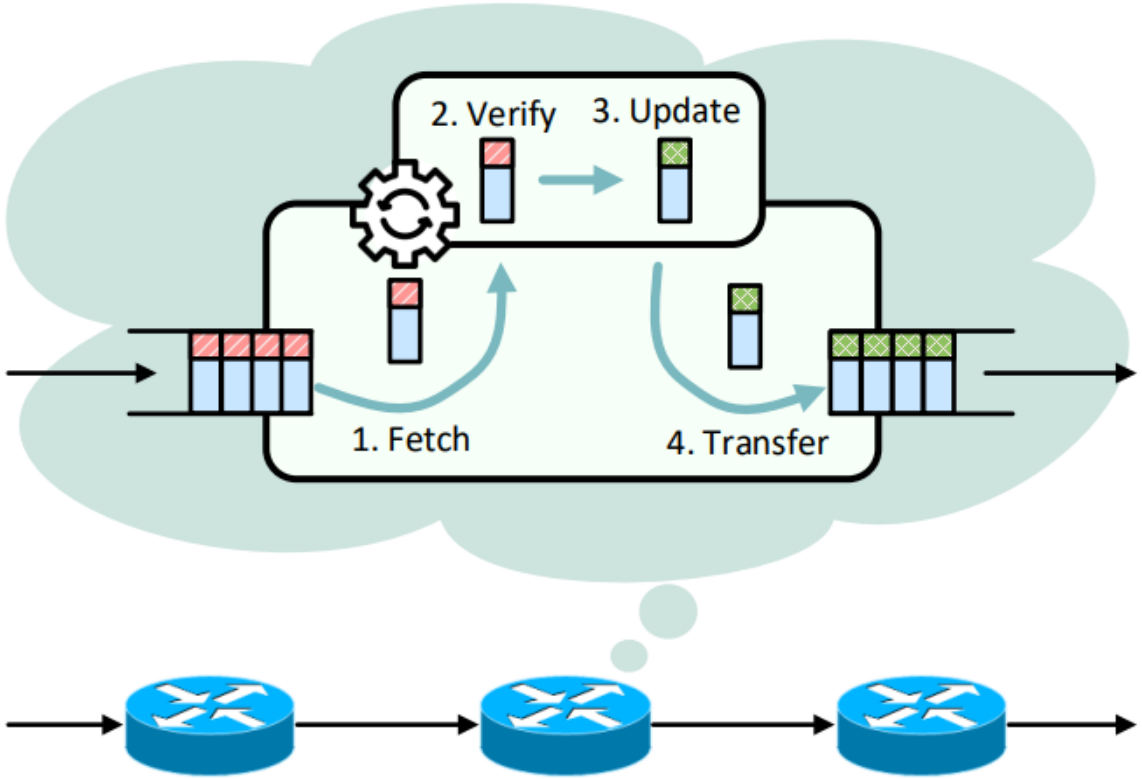
Packet-wise Validation

- Centered on individual packets

Process a single packet at a time

- Constrained by cryptographic techniques

Limit efficiency and security directly because of the adopted cryptographic scheme



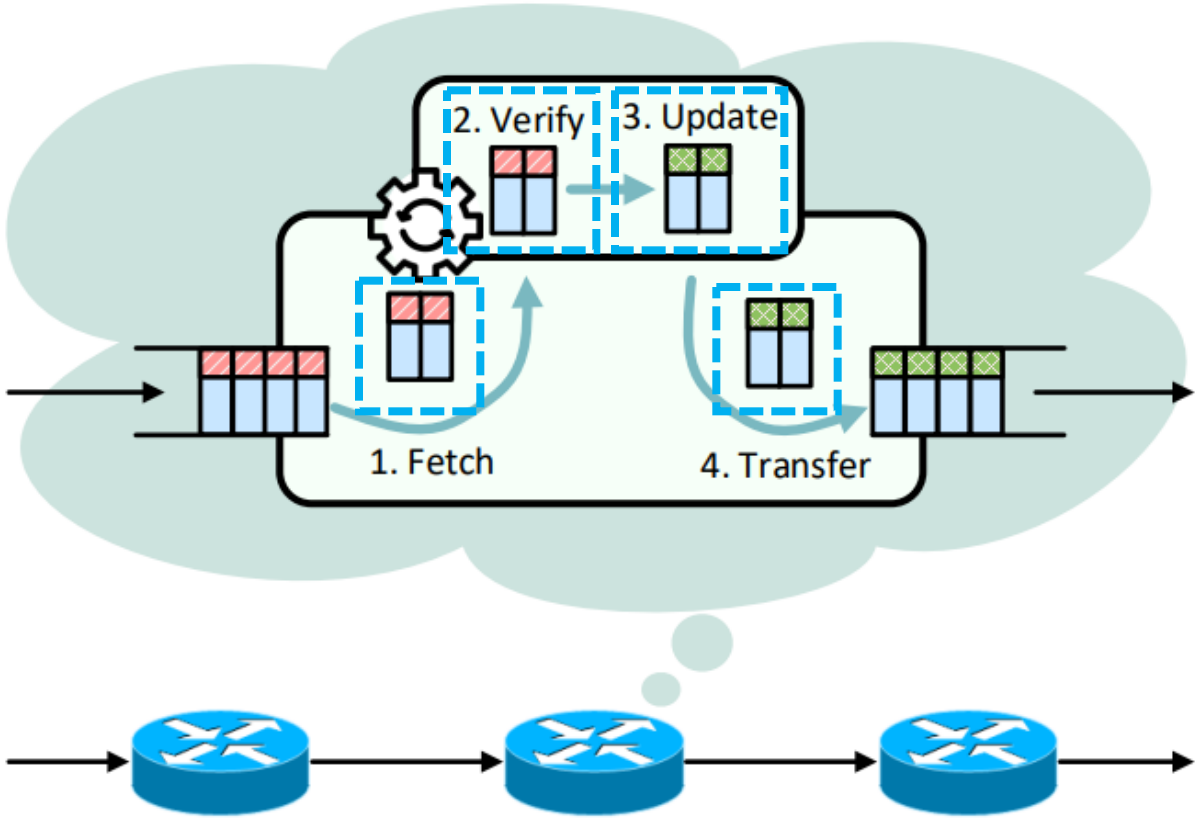
What if we verify a group of packets simultaneously?

Improve efficiency

Guarantee security

…

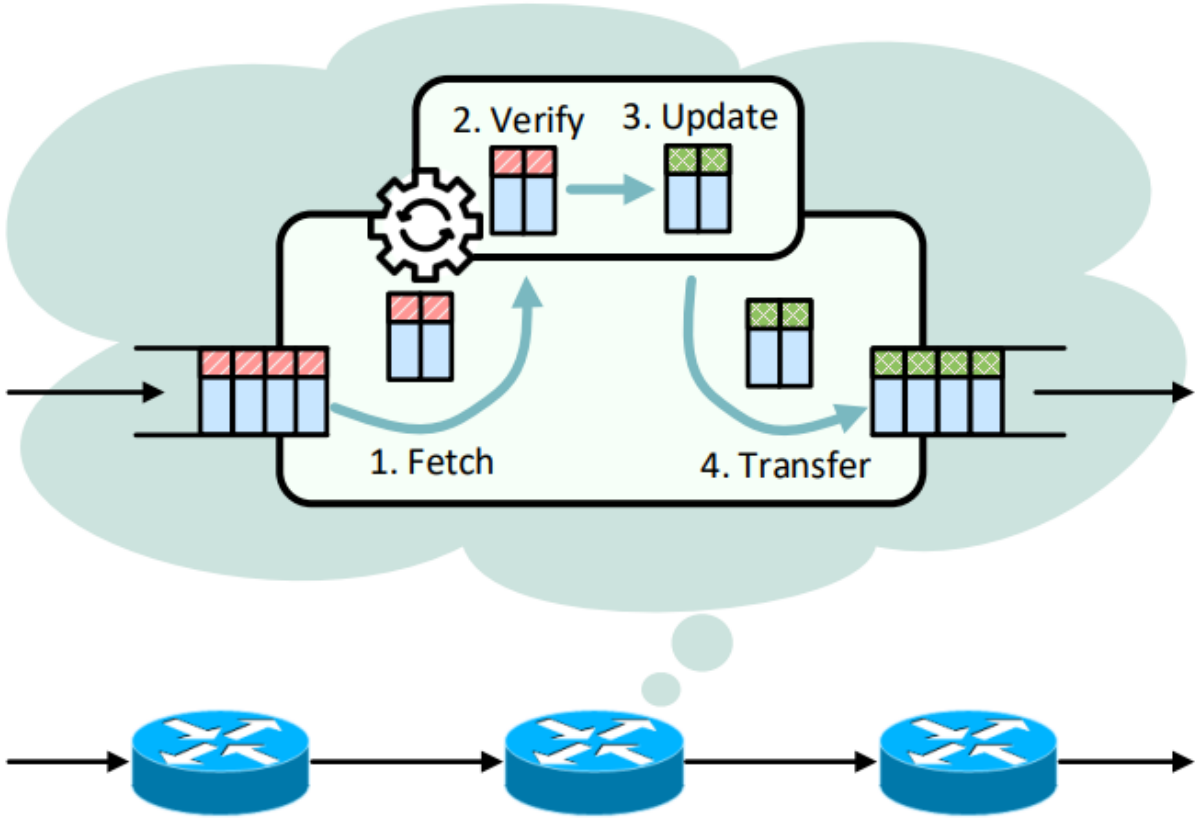
Symphony: Aggregate Validation



routers process packets in a group-wise way

- Aggregate a group of packets as an independent packet
- Amortize the proof and the computation cost evenly over packets
- Assure unchanged security constraints through aggregate validation

Symphony: Aggregate Validation



- The router fetches a group of packets from the input queue
- The router validates the entire group by comparing the computed proof with concatenated proofs in packets
- If validation succeeds, the router updates and evenly distributes its proof across all packets in the group
- The router transfers the updated packet group for output

How to Identify Packet Order

Hierarchical Tags

How to Identify Packet Order

Hierarchical Tags

Group Tag

- Assign packets of the same group with an identical group tag
- Combine SessionID to differentiate groups

How to Identify Packet Order

Hierarchical Tags

Group Tag

Order Tag

- Assign packets of the same group with different order tags

How to Identify Packet Order

Hierarchical Tags

Group Tag

Order Tag

Group Size

- Instruct the exact number of packets in a group
- Identify packet loss
- Make group size adjustable

How to Identify Packet Order

Hierarchical Tags

Group Tag

Order Tag

Group Size

Packet	SessionID	GroupTag	OrderTag	GroupSize
A	$S1$	1	0	3
B	$S1$	1	2	3
C	$S2$	2	2	4
D	$S1$	1	1	3

How to Identify Packet Order

Hierarchical Tags

Group Tag

Order Tag

Group Size

Packet	SessionID	GroupTag	OrderTag	GroupSize
A	$S1$	1	0	3
B	$S1$	1	2	3
C	$S2$	2	2	4
D	$S1$	1	1	3

Queue 1: packet A

How to Identify Packet Order

Hierarchical Tags

Group Tag

Order Tag

Group Size

Packet	SessionID	GroupTag	OrderTag	GroupSize
A	$S1$	1	0	3
B	$S1$	1	2	3
C	$S2$	2	2	4
D	$S1$	1	1	3

Queue 1: packet A, packet B

How to Identify Packet Order

Hierarchical Tags

Group Tag

Order Tag

Group Size

Packet	SessionID	GroupTag	OrderTag	GroupSize
A	$S1$	1	0	3
B	$S1$	1	2	3
C	$S2$	2	2	4
D	$S1$	1	1	3

Queue 1: packet A, packet B
Queue 2: packet C

How to Identify Packet Order

Hierarchical Tags

Group Tag

Order Tag

Group Size

Packet	SessionID	GroupTag	OrderTag	GroupSize
A	$S1$	1	0	3
B	$S1$	1	2	3
C	$S2$	2	2	4
D	$S1$	1	1	3

Queue 1: packet A, packet D, packet B
Queue 2: packet C

How to Deal with Packet Loss

- Straightforward solution: simply drop all the incomplete groups

The evaluation results show that Symphony still has a higher throughput comparing with EPIC when packet loss rate is up to 10%

How to Deal with Packet Loss

- Straightforward solution: simply drop all the incomplete groups
- Packet reaggregation: re-initiate aggregation validation over the remaining packets

How to Deal with Packet Loss

- Straightforward solution: simply drop all the incomplete groups
- Packet reaggregation: re-initiate aggregation validation over the remaining packets

Embed an entire proof seed

The seed is carried by all packets in a group

Routers can use the seed to compute a valid proof

How to Deal with Packet Loss

- Straightforward solution: simply drop all the incomplete groups
- Packet reaggregation: re-initiate aggregation validation over the remaining packets

Embed an entire proof seed

Verify the remaining packets

Routers compare the computed proof with the remaining segments of proofs

How to Deal with Packet Loss

- Straightforward solution: simply drop all the incomplete groups
- Packet reaggregation: re-initiate aggregation validation over the remaining packets

Embed an entire proof seed

Verify the remaining packets

Generate new proofs to replace the old

Routers use the left packets as an input to compute new path validation proofs

How to Deal with Packet Loss

- Straightforward solution: simply drop all the incomplete groups
- Packet reaggregation: re-initiate aggregation validation over the remaining packets

Embed an entire proof seed

Verify the remaining packets

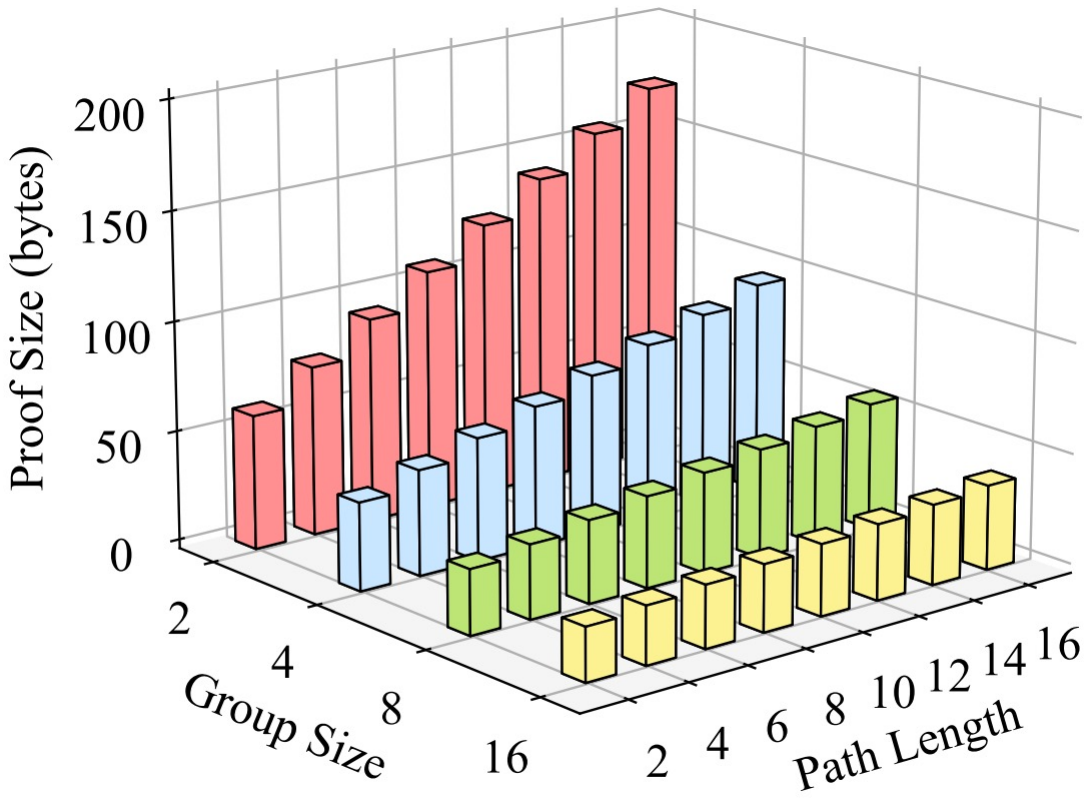
Generate new proofs to replace the old

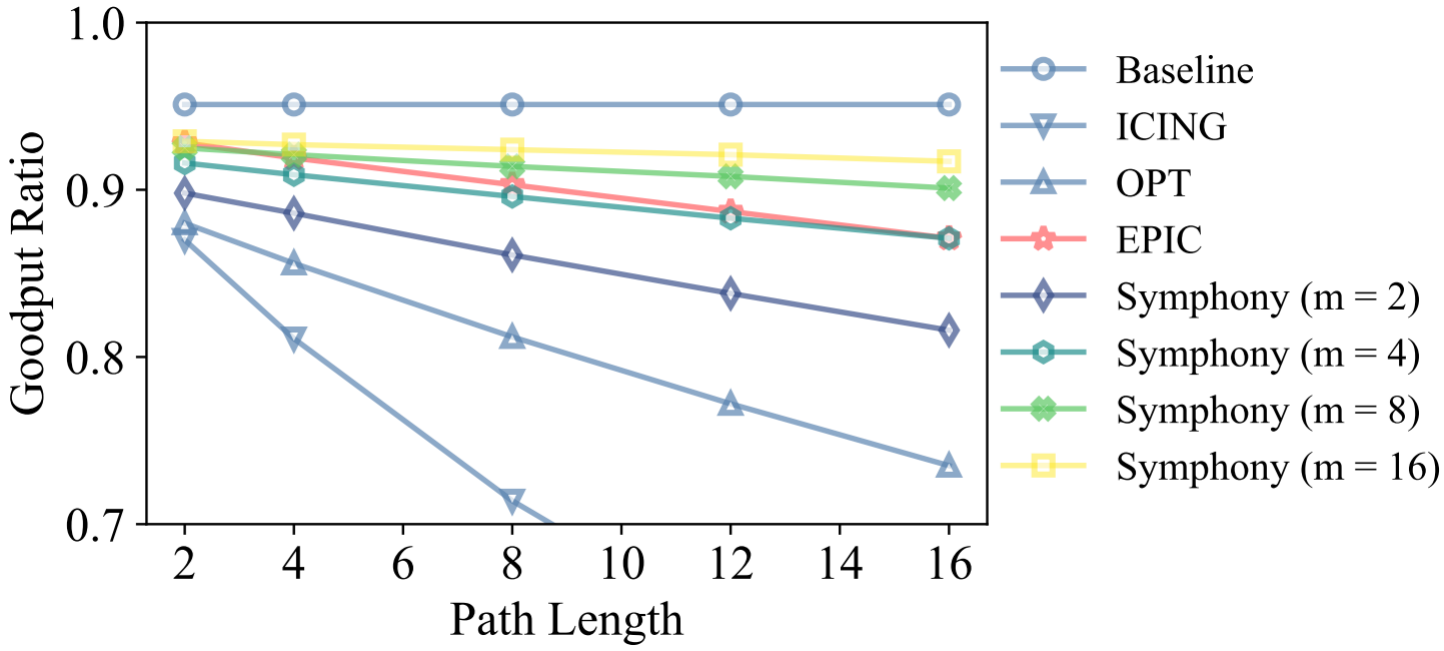
Update a source-generated ciphertext

Routers use their own keys to update the encrypted packet message

Prevent packets from being modified by malicious routers

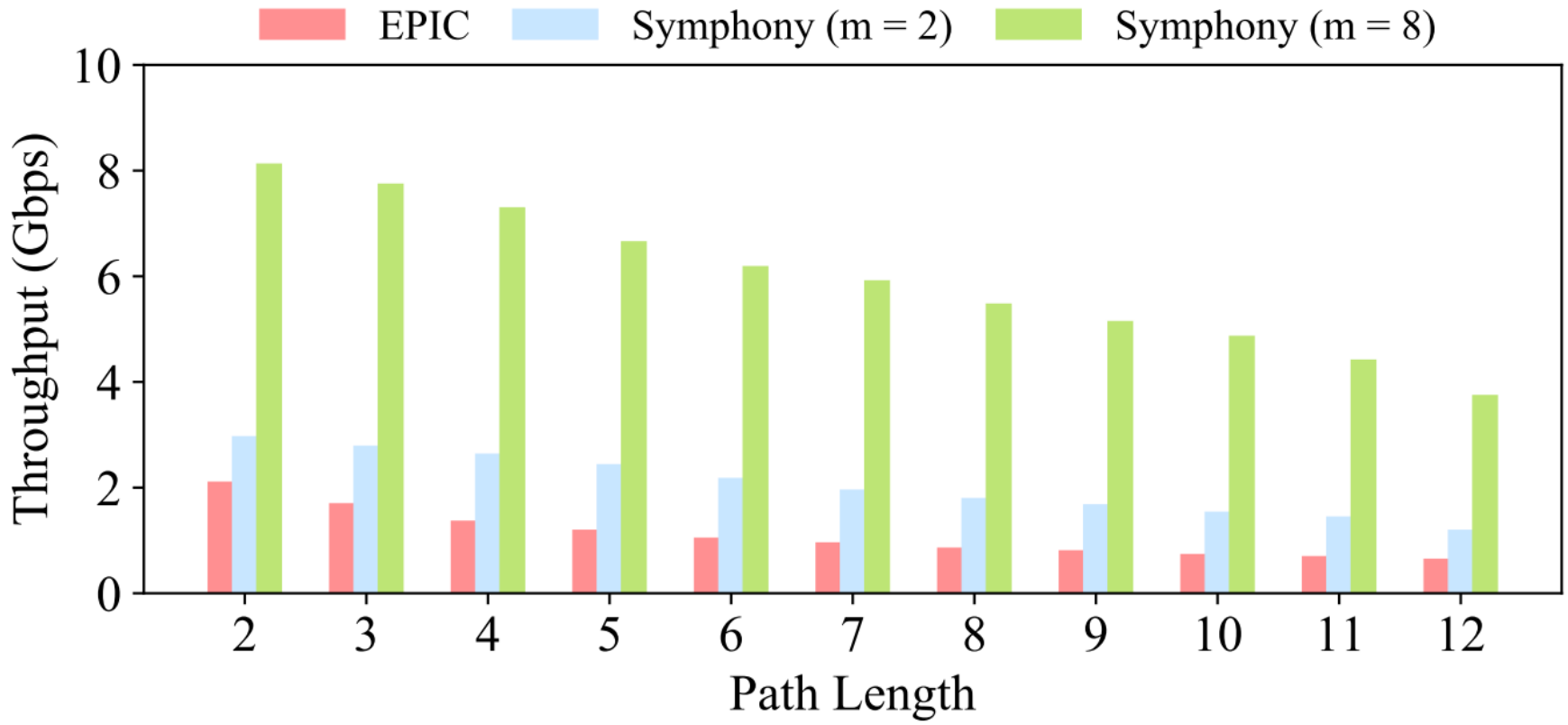
Evaluation: Low Communication Overhead





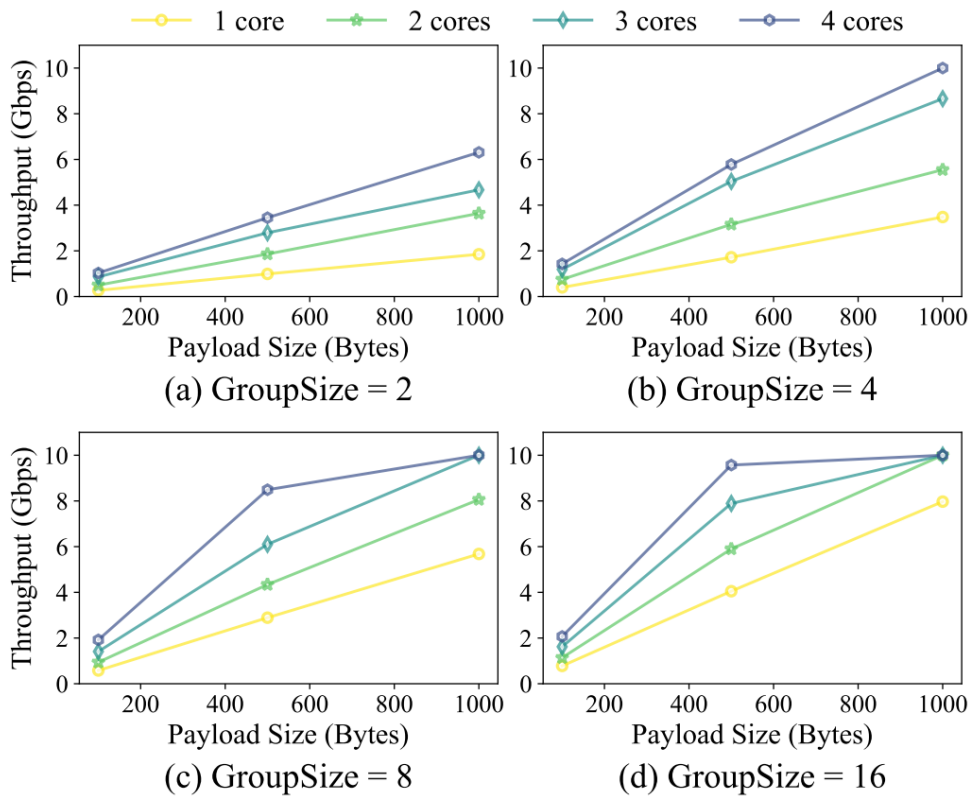
Symphony increases 5% goodput ratio in comparison with EPIC

Evaluation: High Efficiency



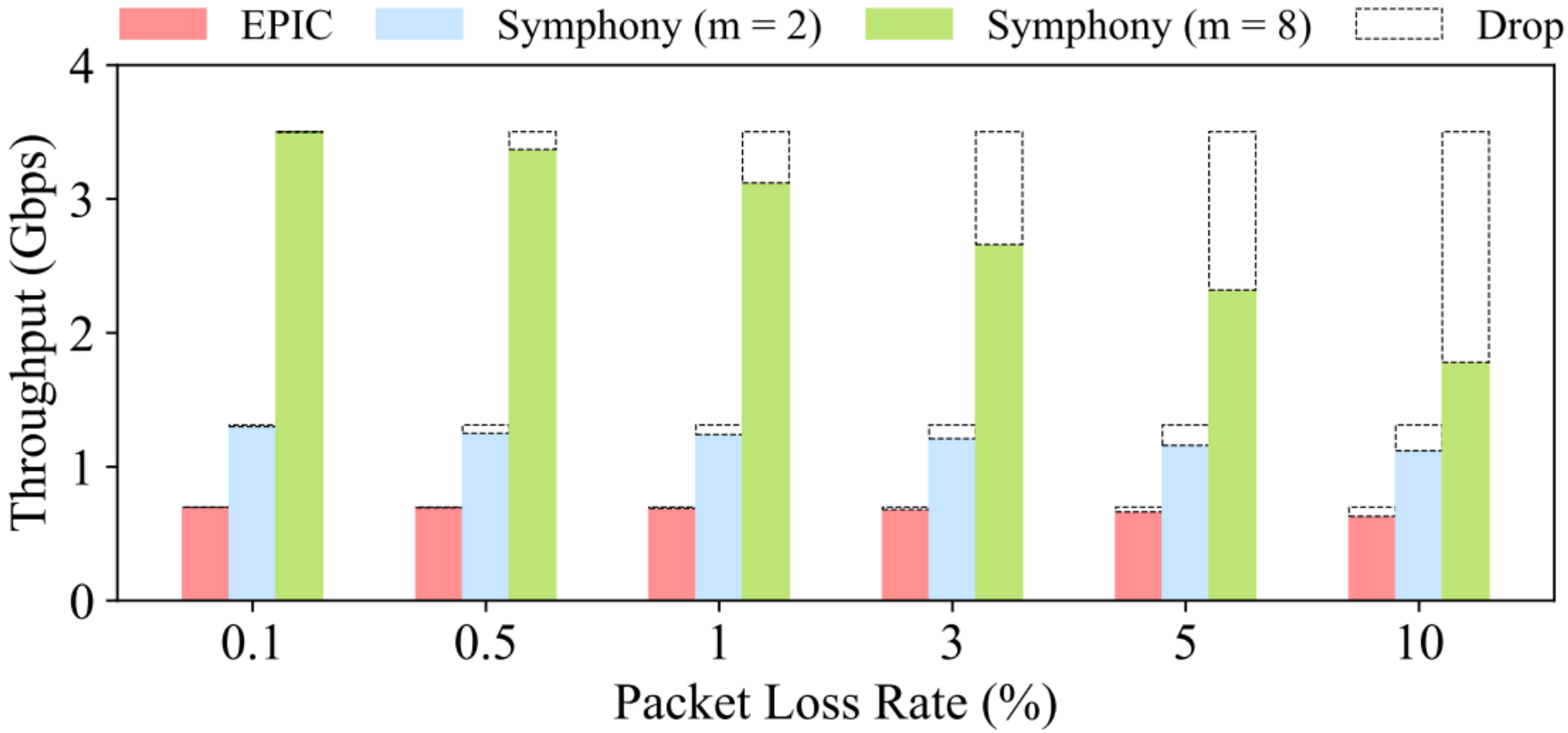
Symphony achieves 1.41×~5.84× higher throughput than EPIC does

Evaluation: High Parallelizability



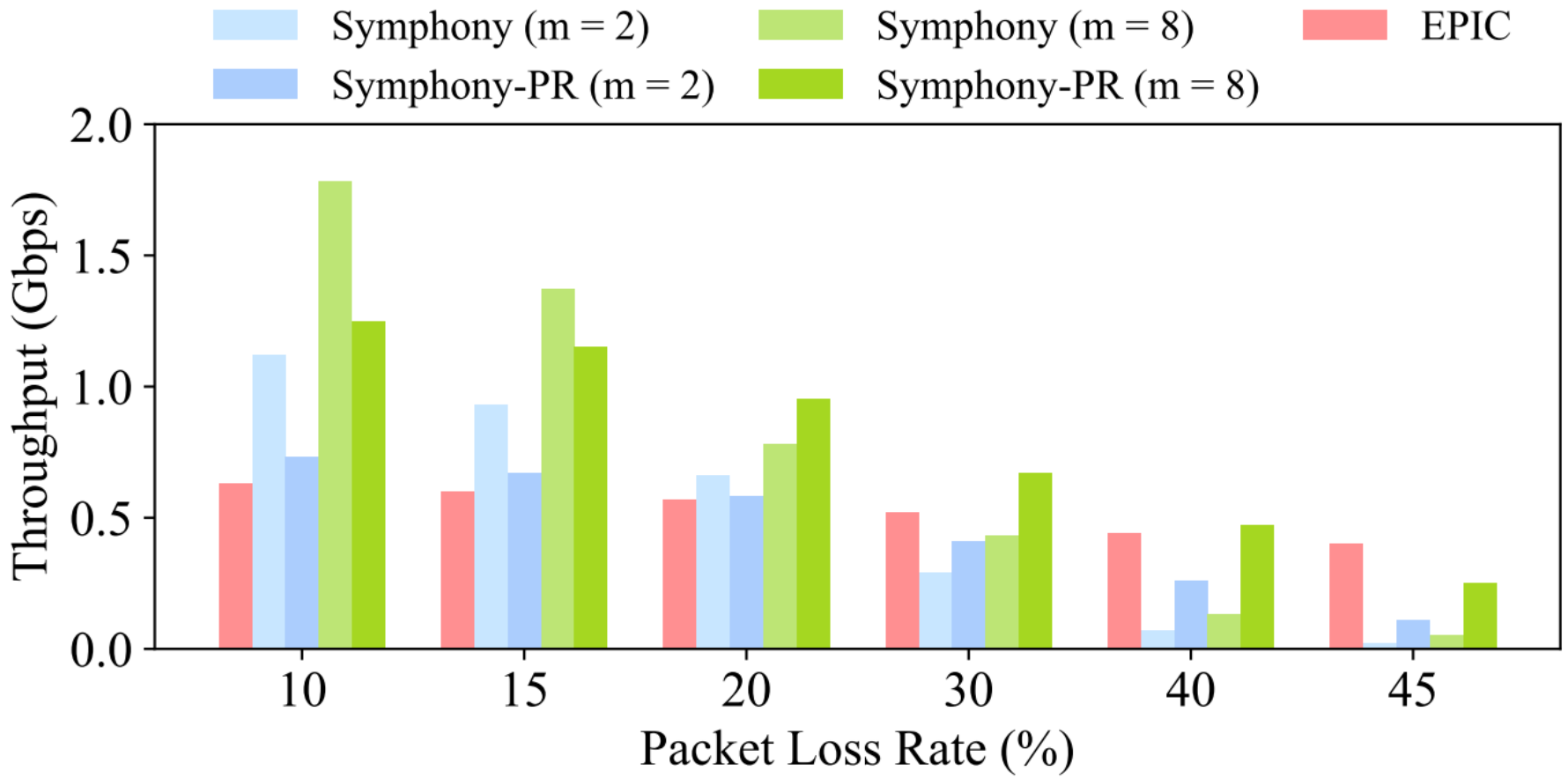
Symphony reaches the upper bound of 10 Gbps link using 3 cores

Evaluation: Resistance to Packet Loss



Symphony outperforms EPIC with a 1.5× higher throughput

Evaluation: Resistance to Packet Loss



Symphony-PR yields a 5× higher throughput than Symphony does given a 45% packet loss rate

More in Paper

- Security analysis

Proof unforgeability

Hop-wise validation

DDoS resistance

- Performance evaluation

Proof processing time

Reassembly time

Reordering time

Mixed packet size

Conclusion

- We amortize validation overhead to improve efficiency yet without sacrificing security
- We propose various techniques to achieve correct and efficient packet aggregation and implement them through Symphony
- We further propose a packet reaggregation technique to efficiently handle packet losses and integrate it into Symphony-PR

Thanks!

If you have any questions about this paper, welcome to contact zjuhax@zju.edu.cn