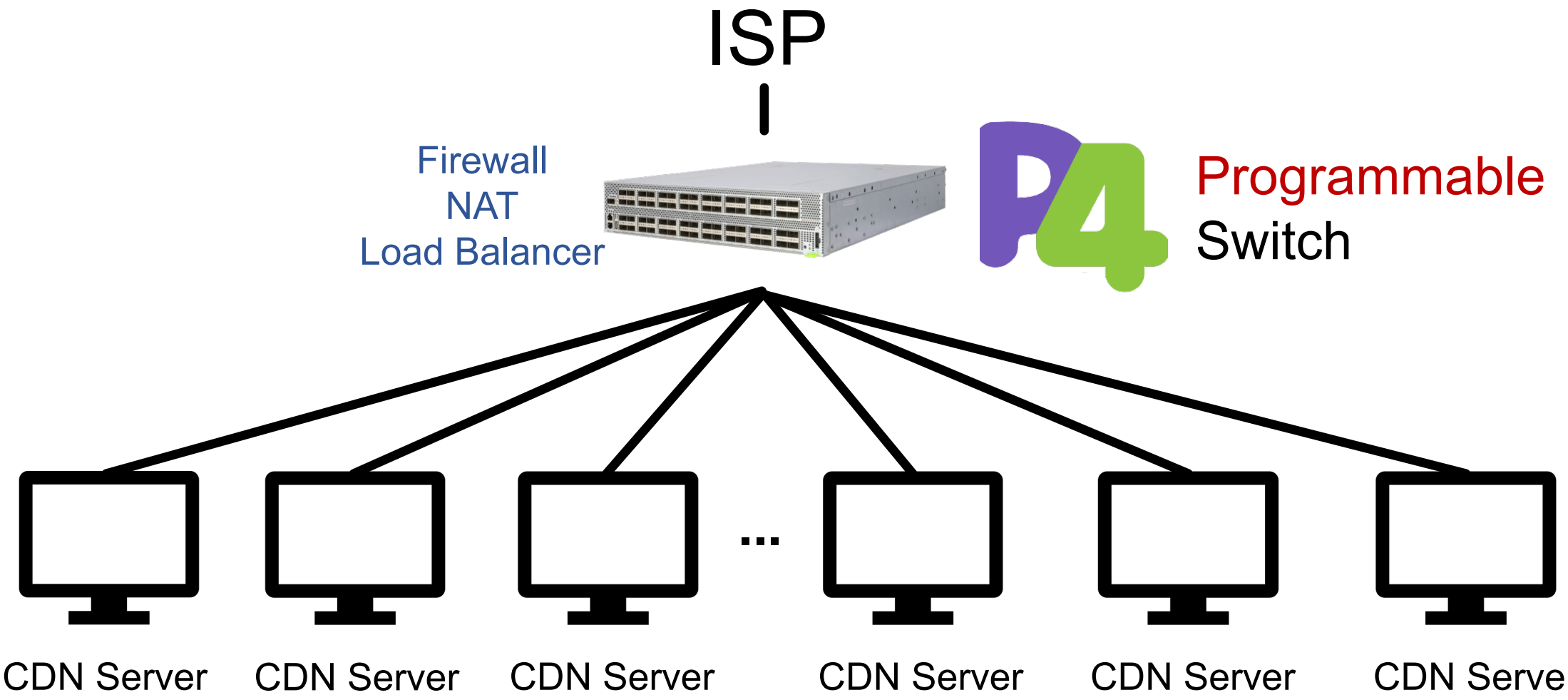
SIGCOMM 2022
Amsterdam

Meissa: Scalable Network Testing for Programmable Data Planes

Naiqian Zheng, Mengqi Liu, Ennan Zhai, Hongqiang Harry Liu,
Yifan Li, Kaicheng Yang, Xuanzhe Liu, Xin Jin

北京大学
PEKING UNIVERSITY

Programmable switches in networks



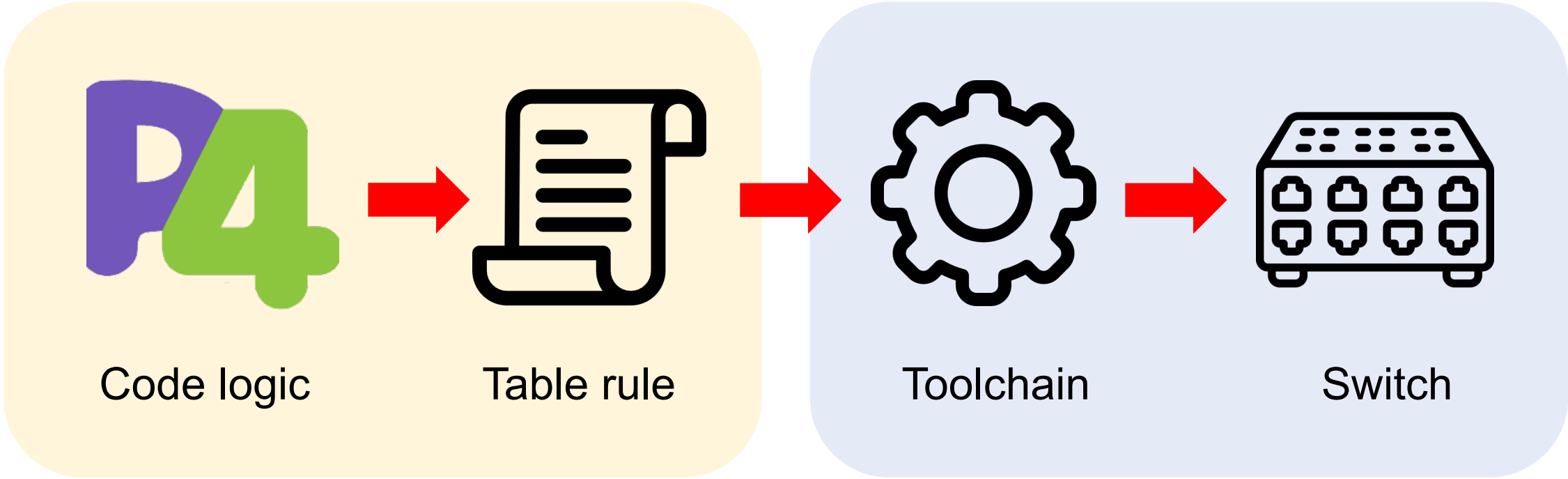
Programmable data planes are buggy

- ✓ Versatility
- ✓ High performance
- ✓ Programmability

- ! Large control flow
- ! Non-code bugs
- ! Incomprehensive test

✗ Bugs are common with programmable data planes!

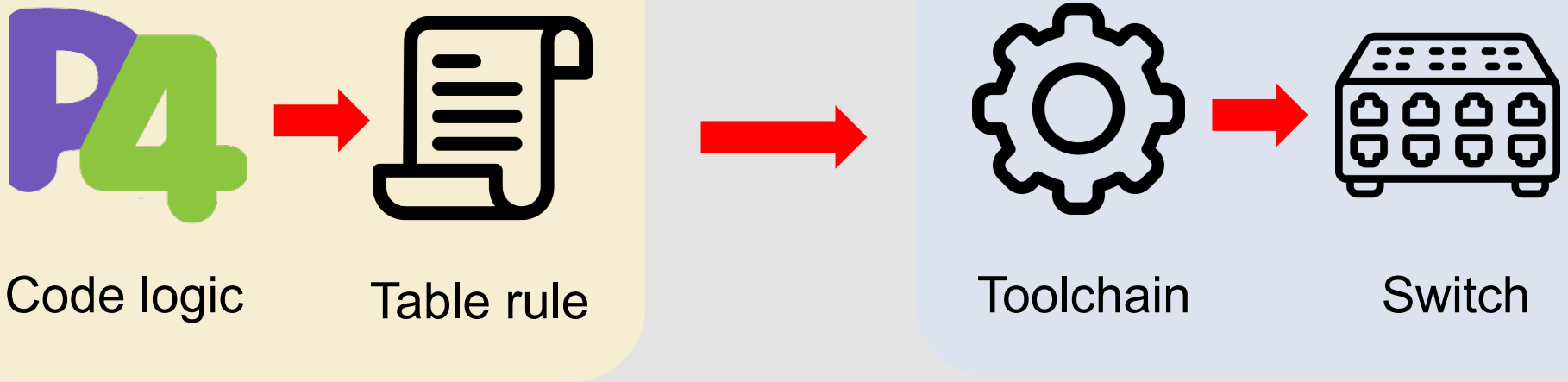
Bug taxonomy



Tools to identify bugs

Testing: Gauntlet, p4pktgen

Verification: Aquila, p4v

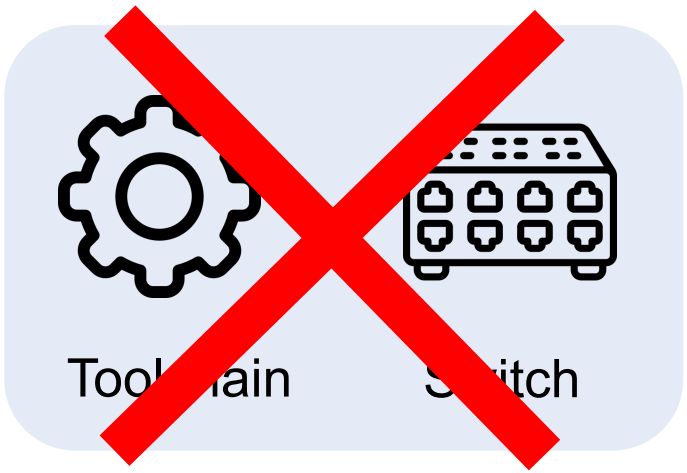


Code bugs

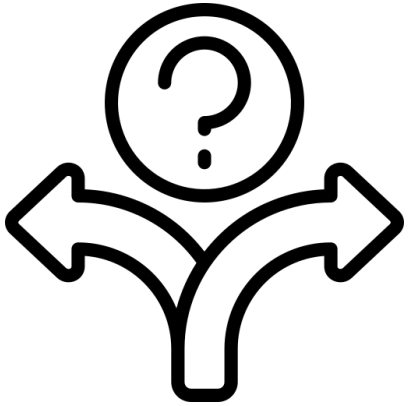
Non-code bugs

Challenge

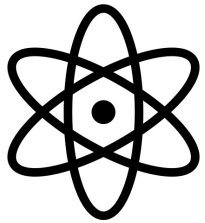
Verification: Aquila, p4v



Non-code bugs



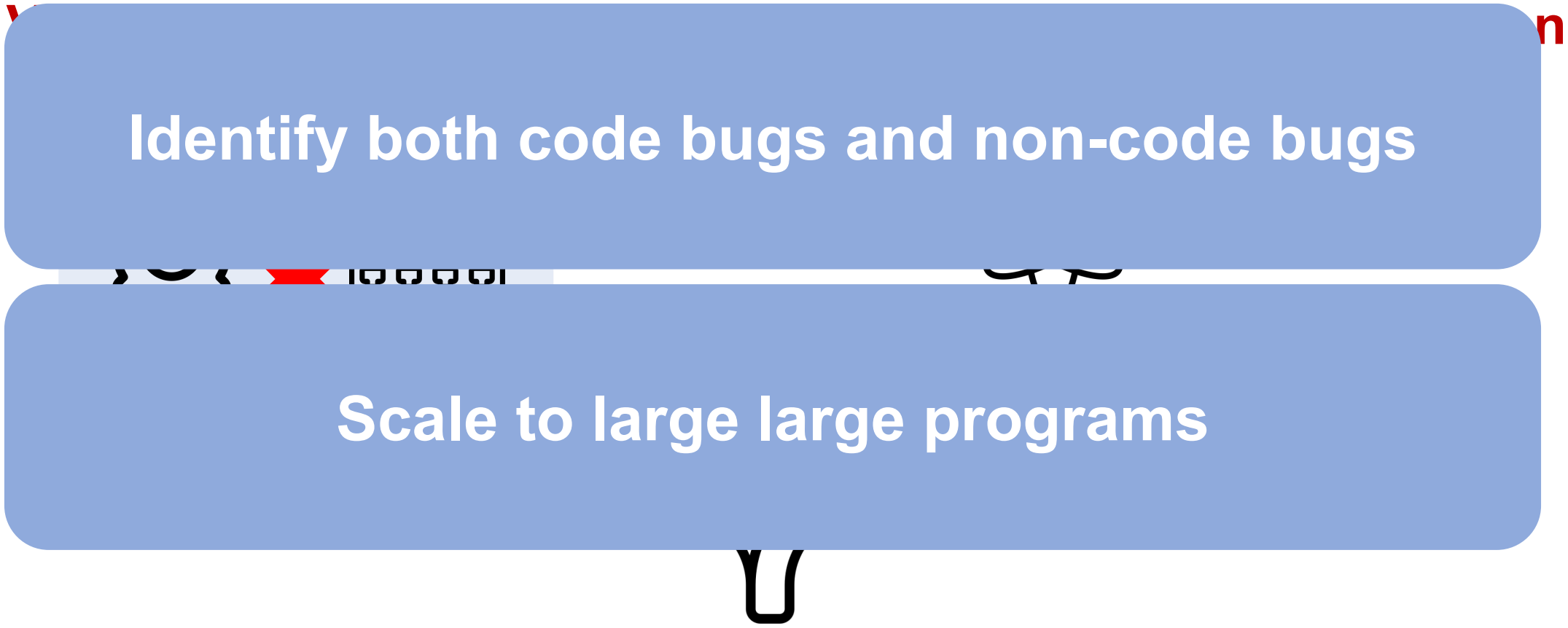
Testing: Gauntlet, p4pktgen



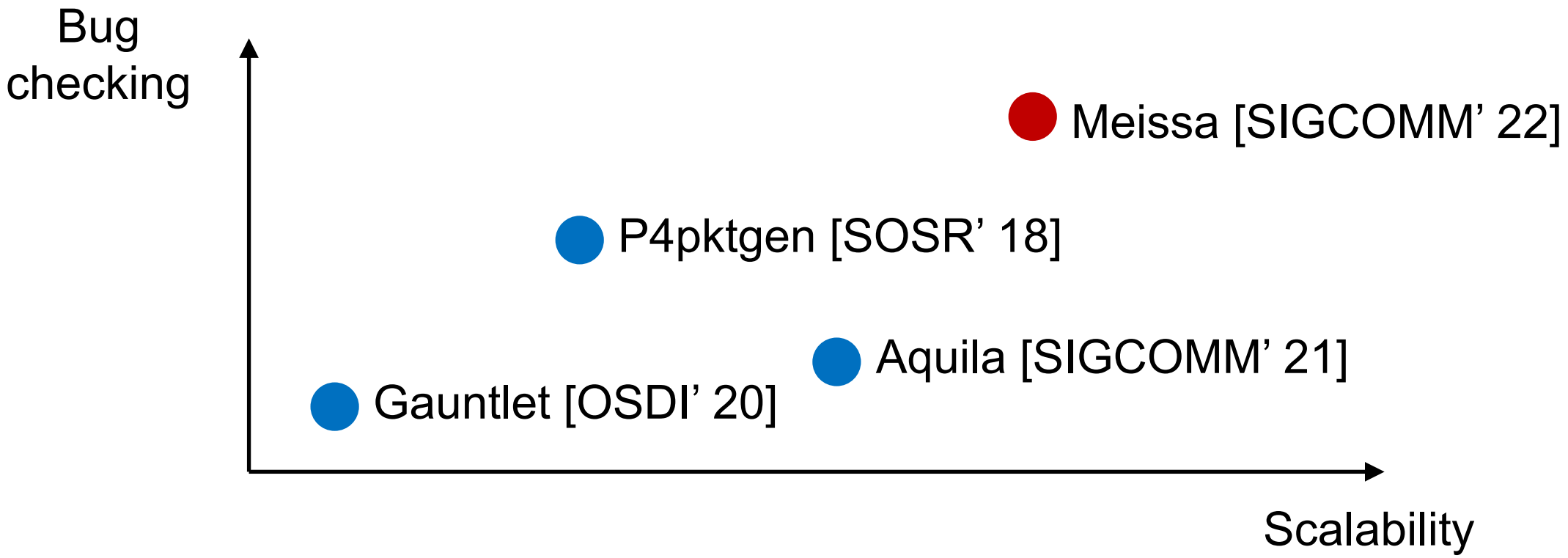
Path Explosion

LOC: $O(10^4)$
# of path: $O(10^{197})$

Challenge

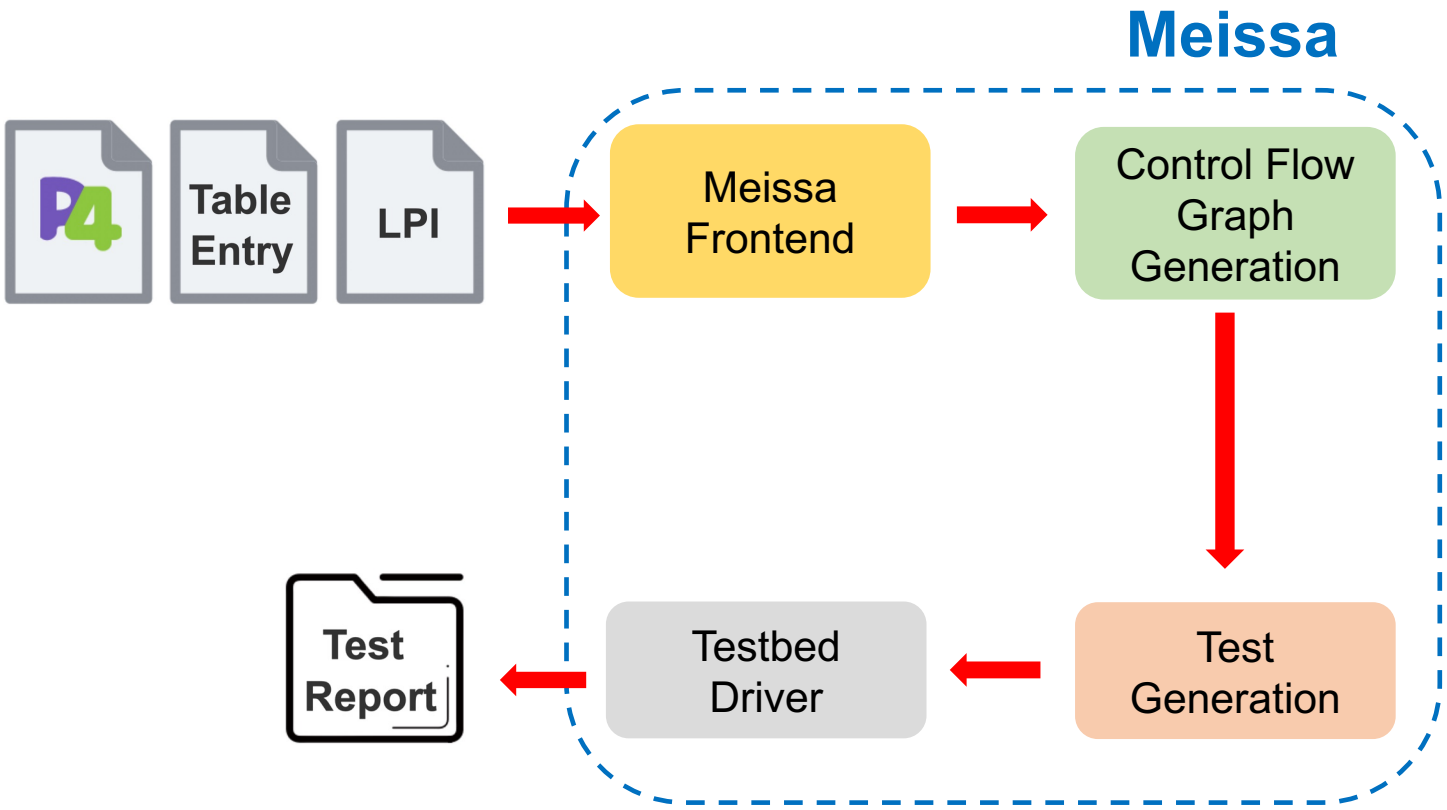

Identify both code bugs and non-code bugs
Scale to large large programs

How to detect potential bugs?

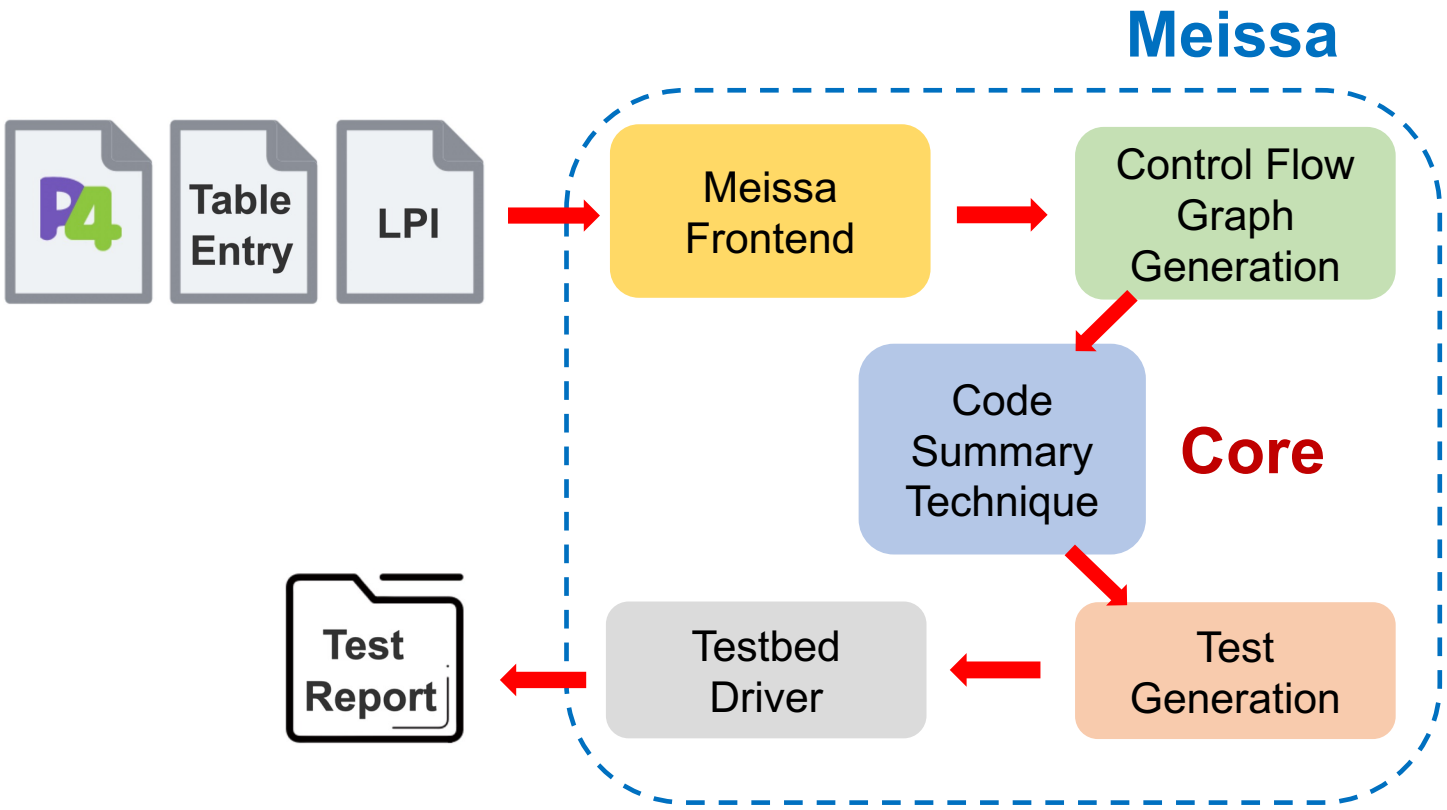


Scalable testing with 100% path coverage

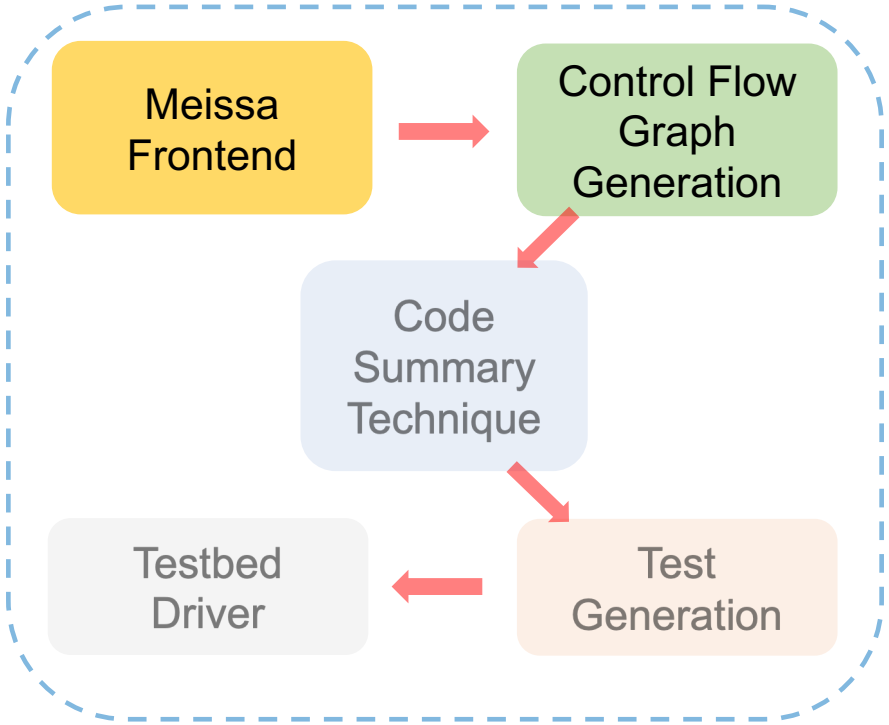
Meissa overview

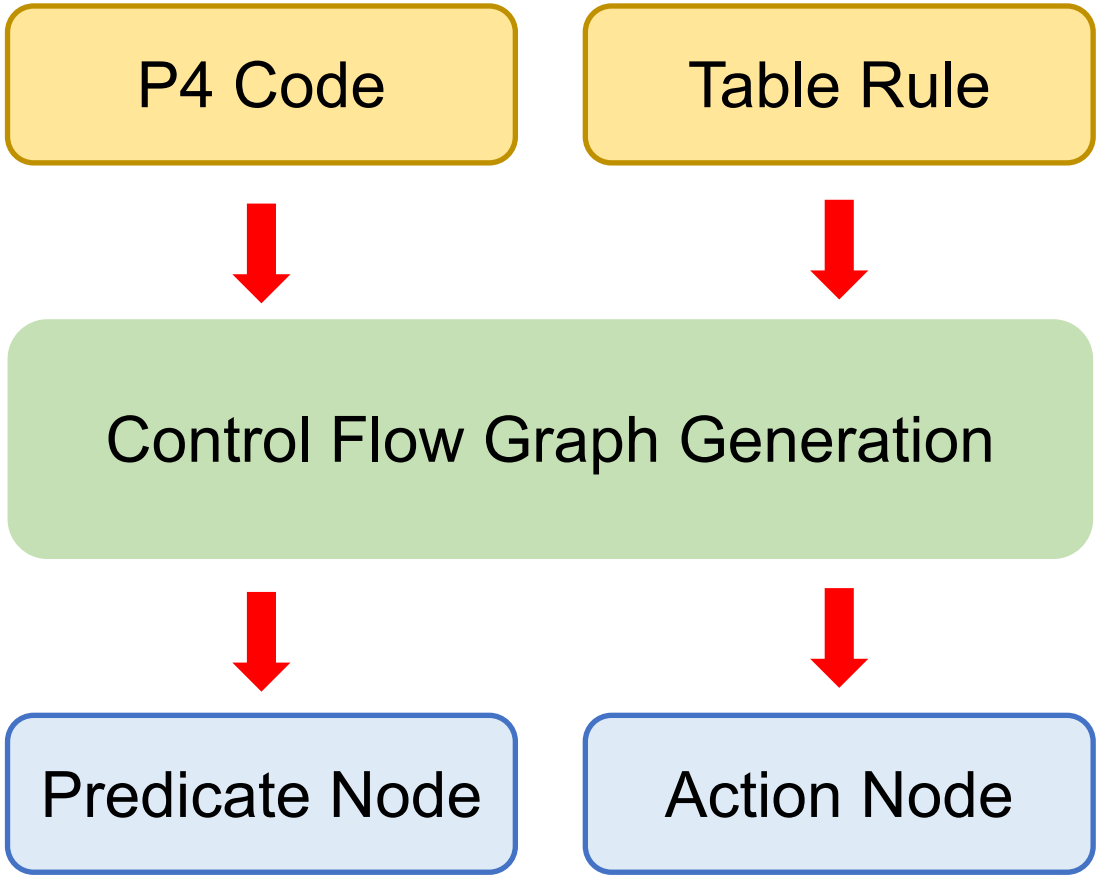


Meissa overview

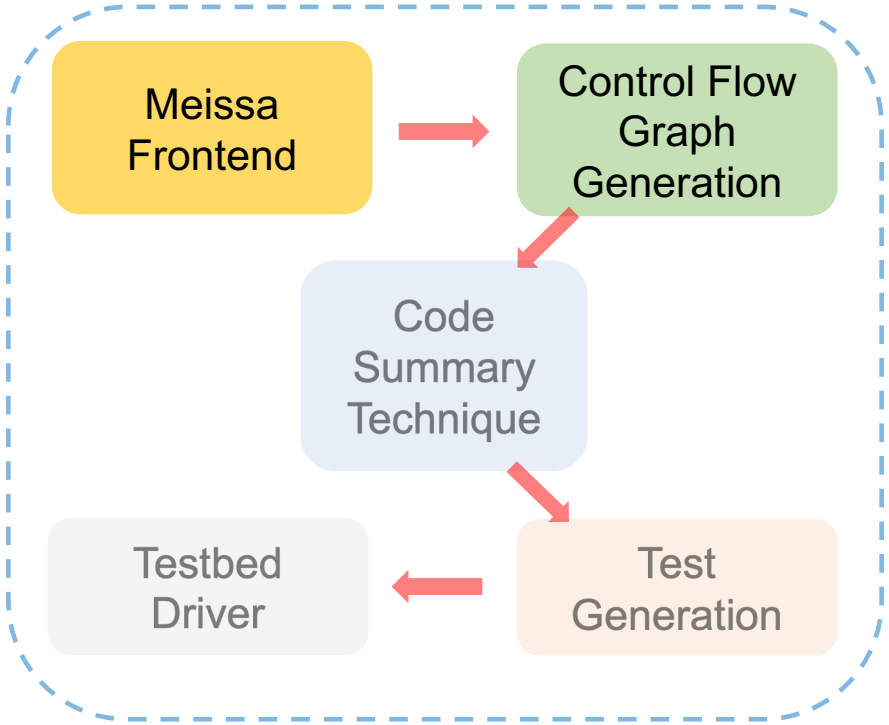


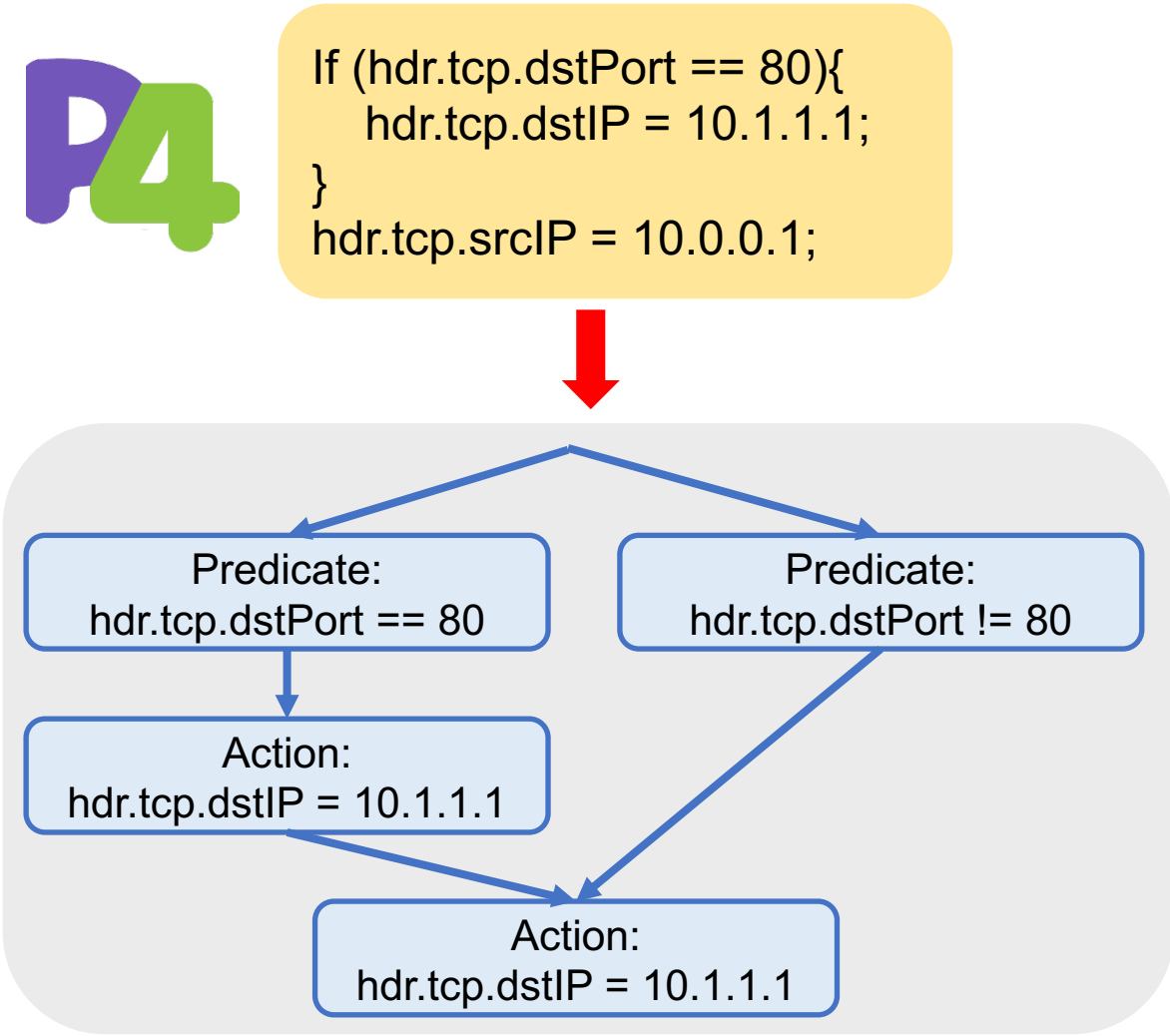
Control flow graph



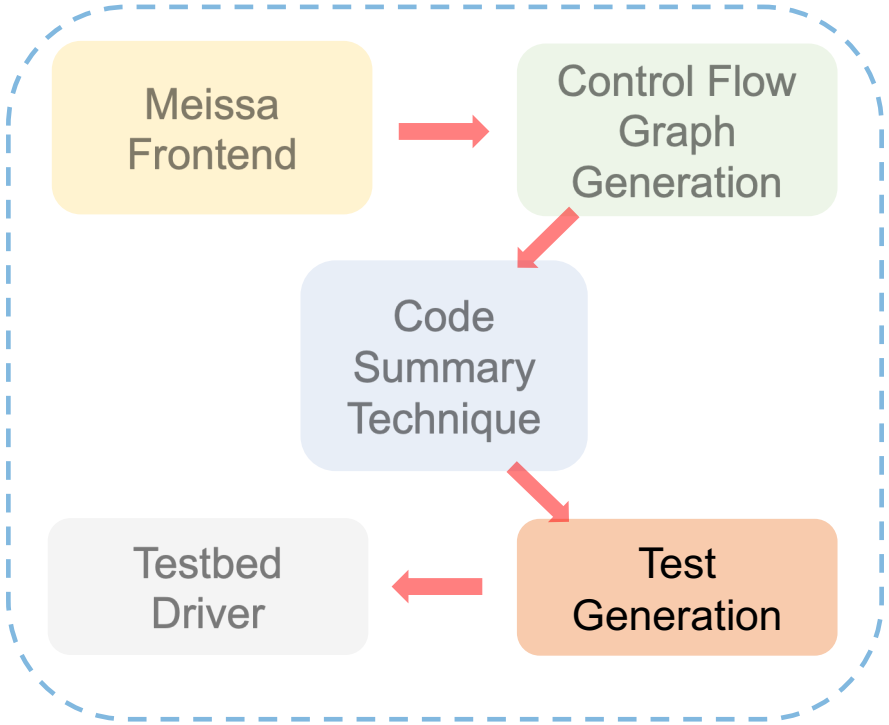


Control flow graph





Test generation

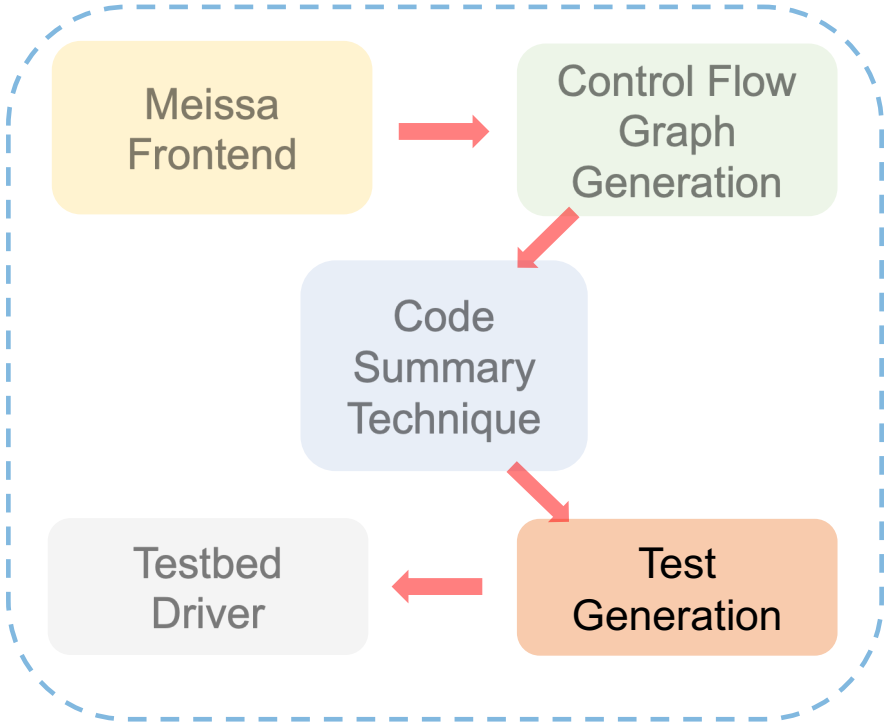


Goal: get input packets which traverse **all** paths in the control flow graph.

Depth-first search traverses the control flow graph.

Symbolic execution checks the paths' satisfiability.

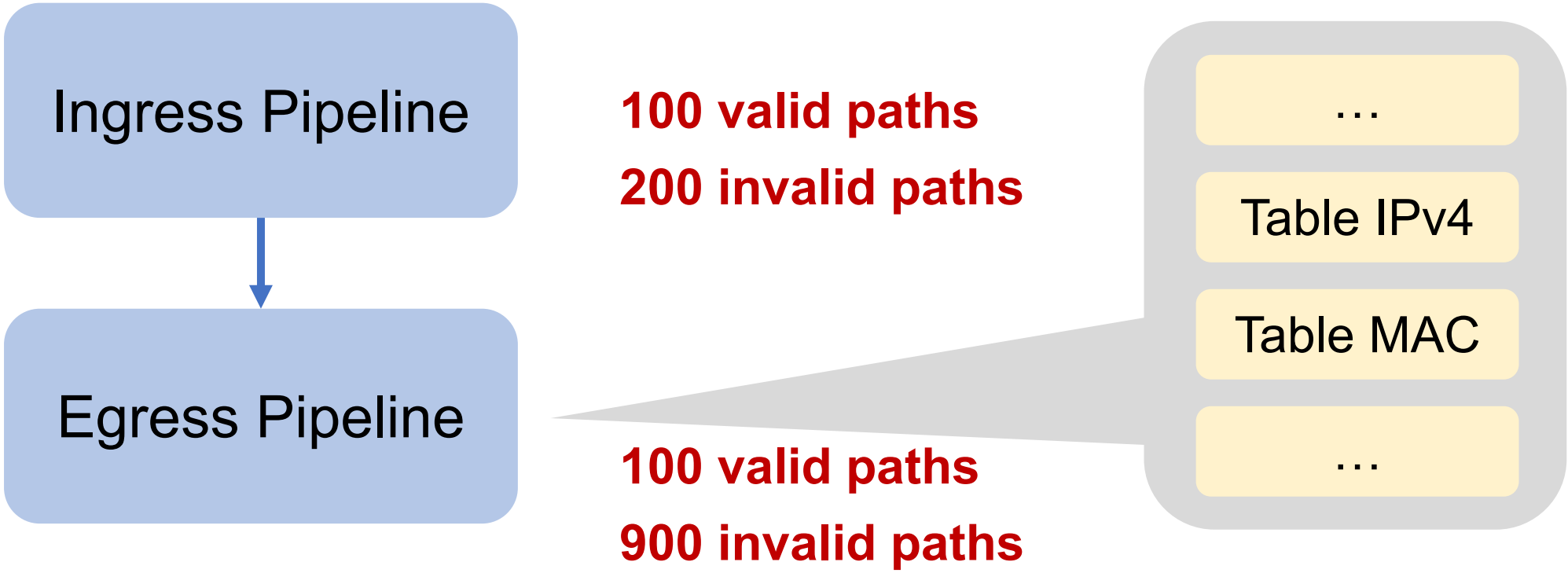
Test generation



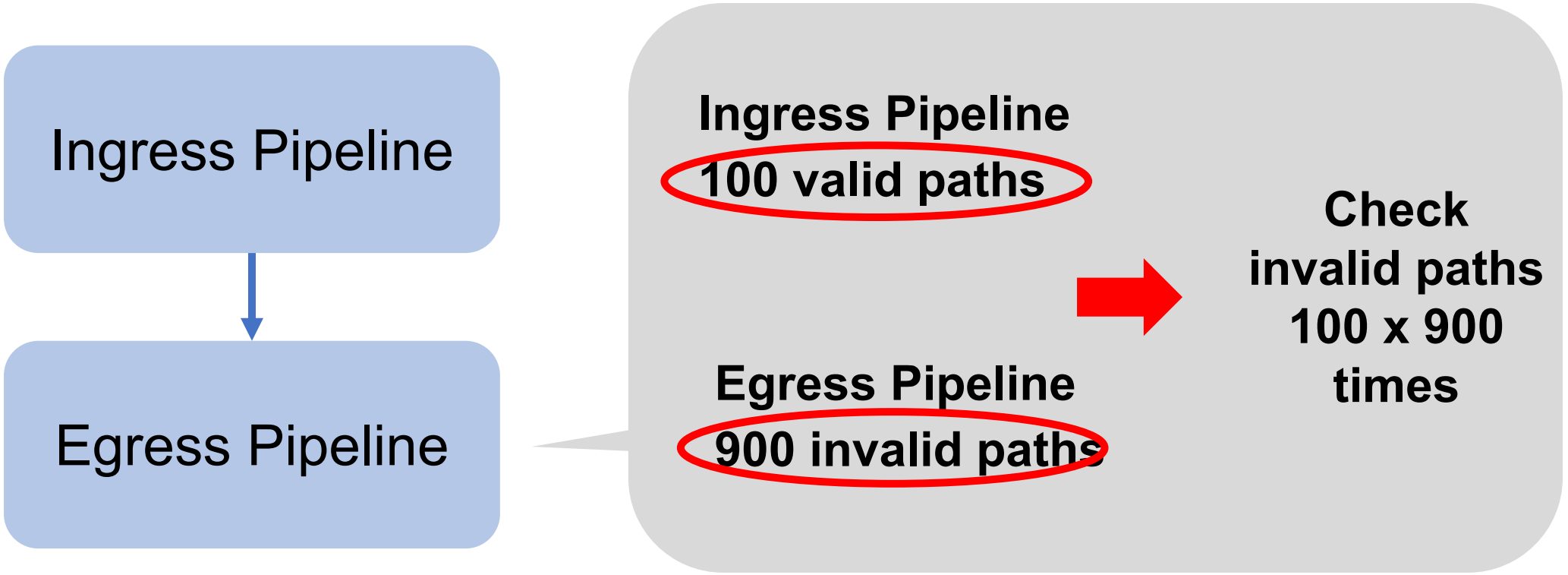
Goal: get input packets which traverse **all** paths in the control flow graph.

Without code summary, scaling to large programs is hard!

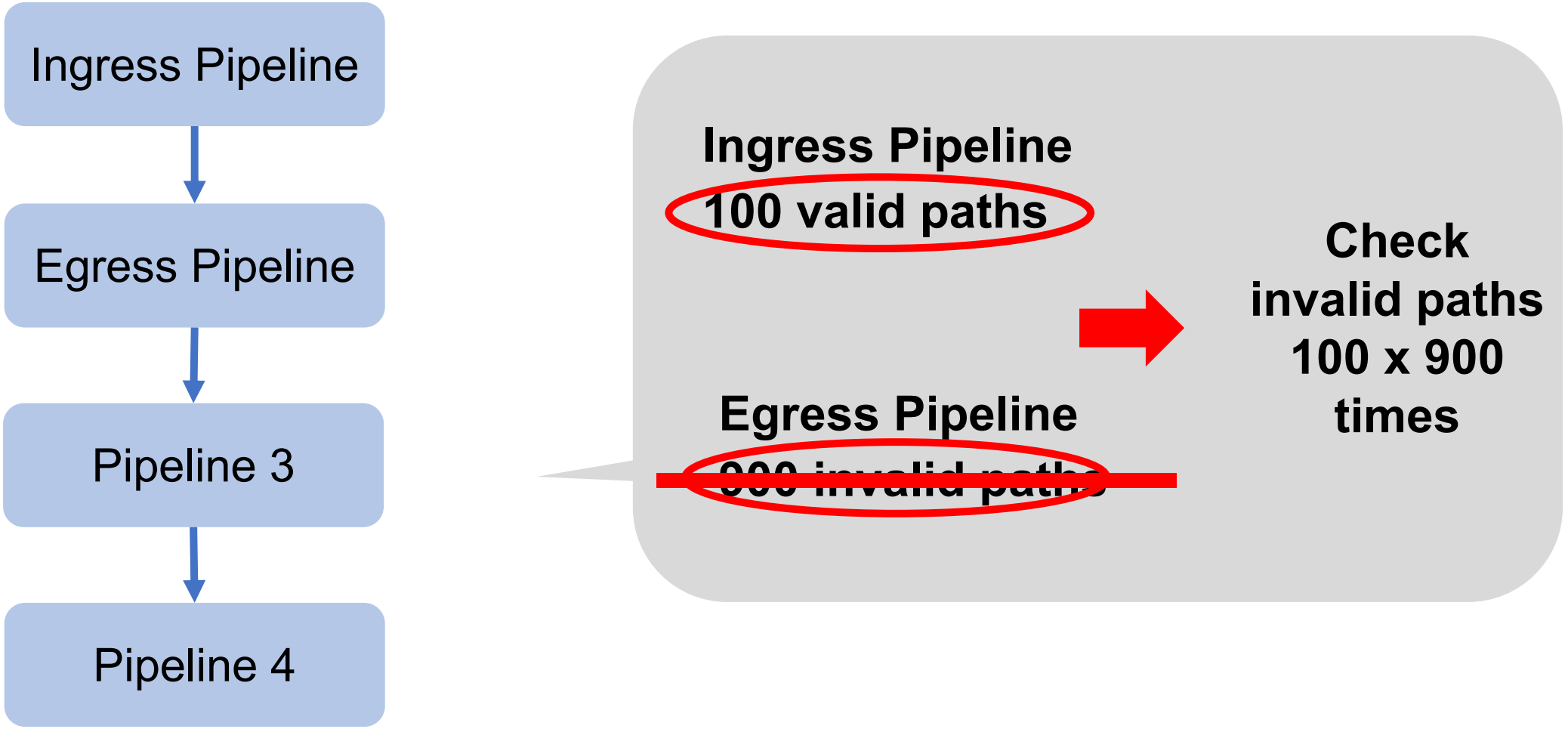
Redundancy


Ingress Pipeline
Egress Pipeline
100 valid paths
200 invalid paths
100 valid paths
900 invalid paths
…
Table IPv4
Table MAC
…

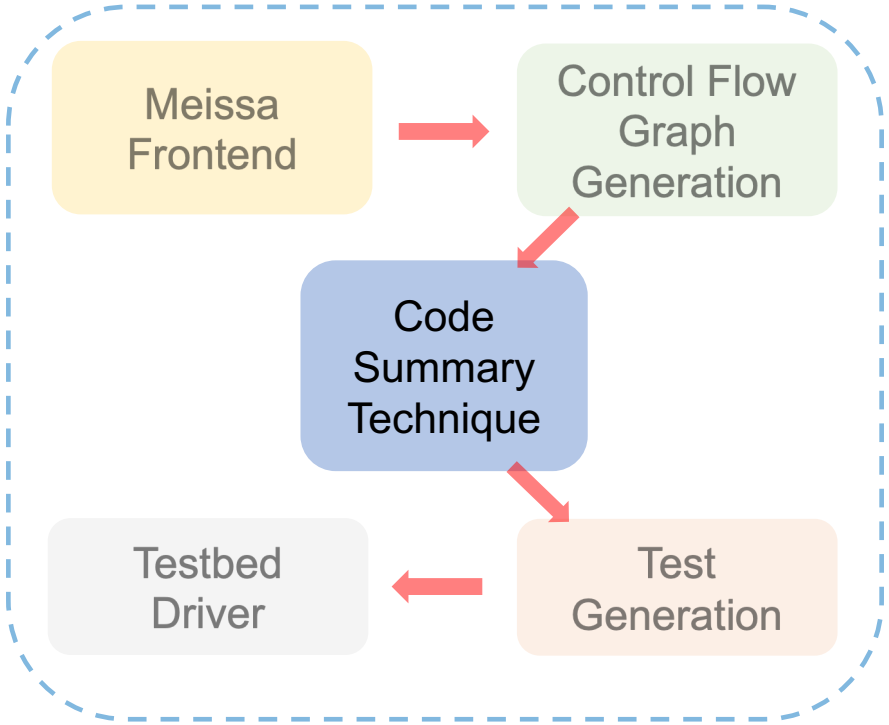
Redundancy


Ingress Pipeline
Egress Pipeline
Ingress Pipeline
100 valid paths
Egress Pipeline
900 invalid paths
Check
invalid paths
100 x 900
times

Redundancy


Ingress Pipeline
Egress Pipeline
Pipeline 3
Pipeline 4
Ingress Pipeline
100 valid paths
Egress Pipeline
000 invalid paths
Check
invalid paths
100 x 900
times

Code summary technique



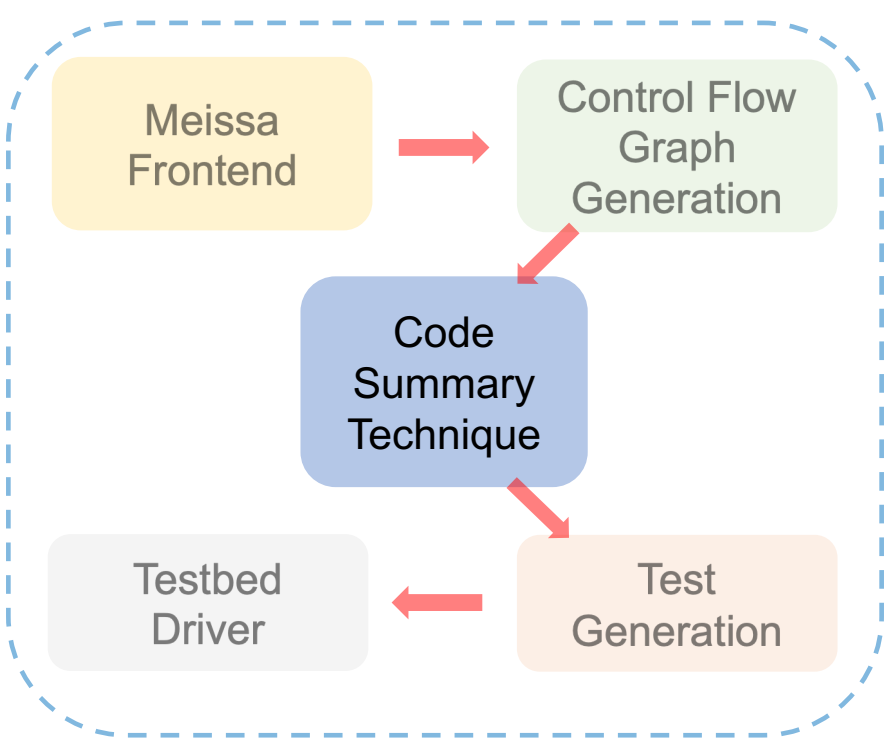
Decompose control flow graph into individual pipelines

Simplify code logics by removing unsatisfiable paths

Update control flow graph with simplified logics

Iteration over pipelines

Code summary technique



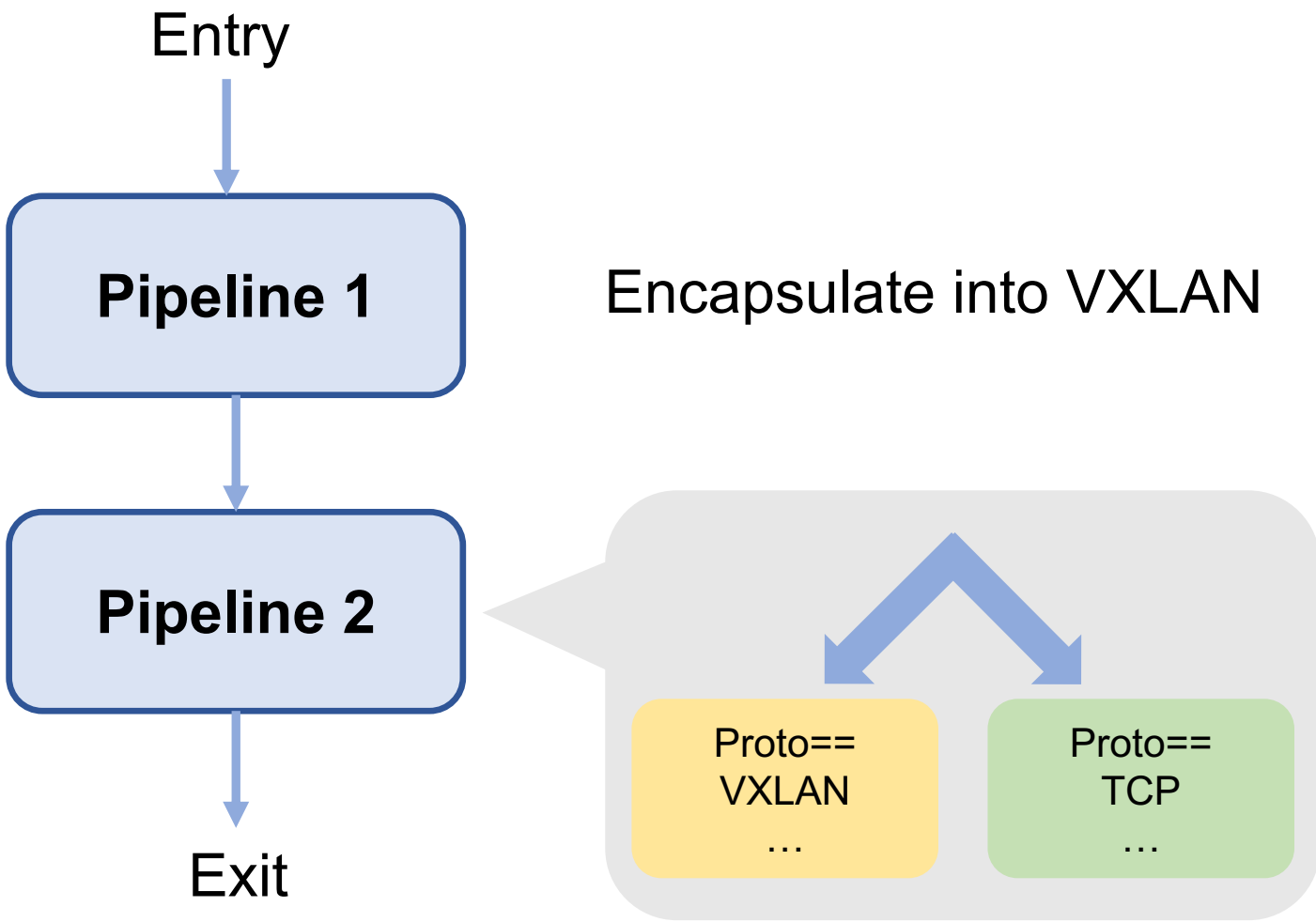
General-purpose languages:
vertical, hard to simplify

P4 languages:
horizontal, significantly simplified

Iteration over pipelines

Pipeline simplification

Simplify code logics by removing unsatisfiable paths



Intra-pipeline redundancy elimination

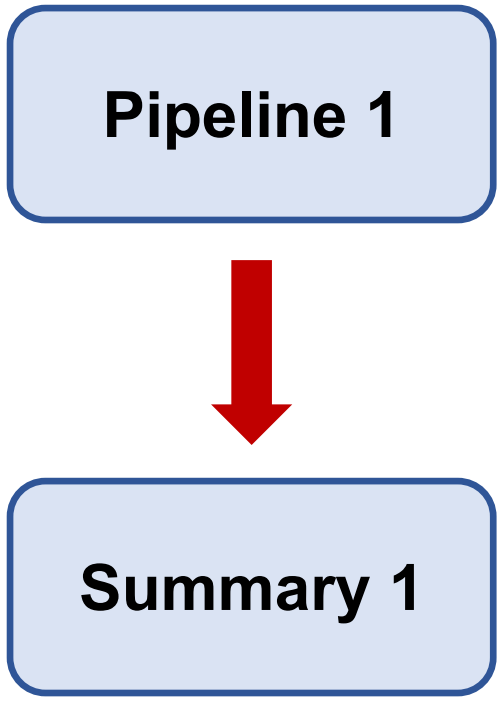
Simplify code logics by removing unsatisfiable paths

Remove invalid paths

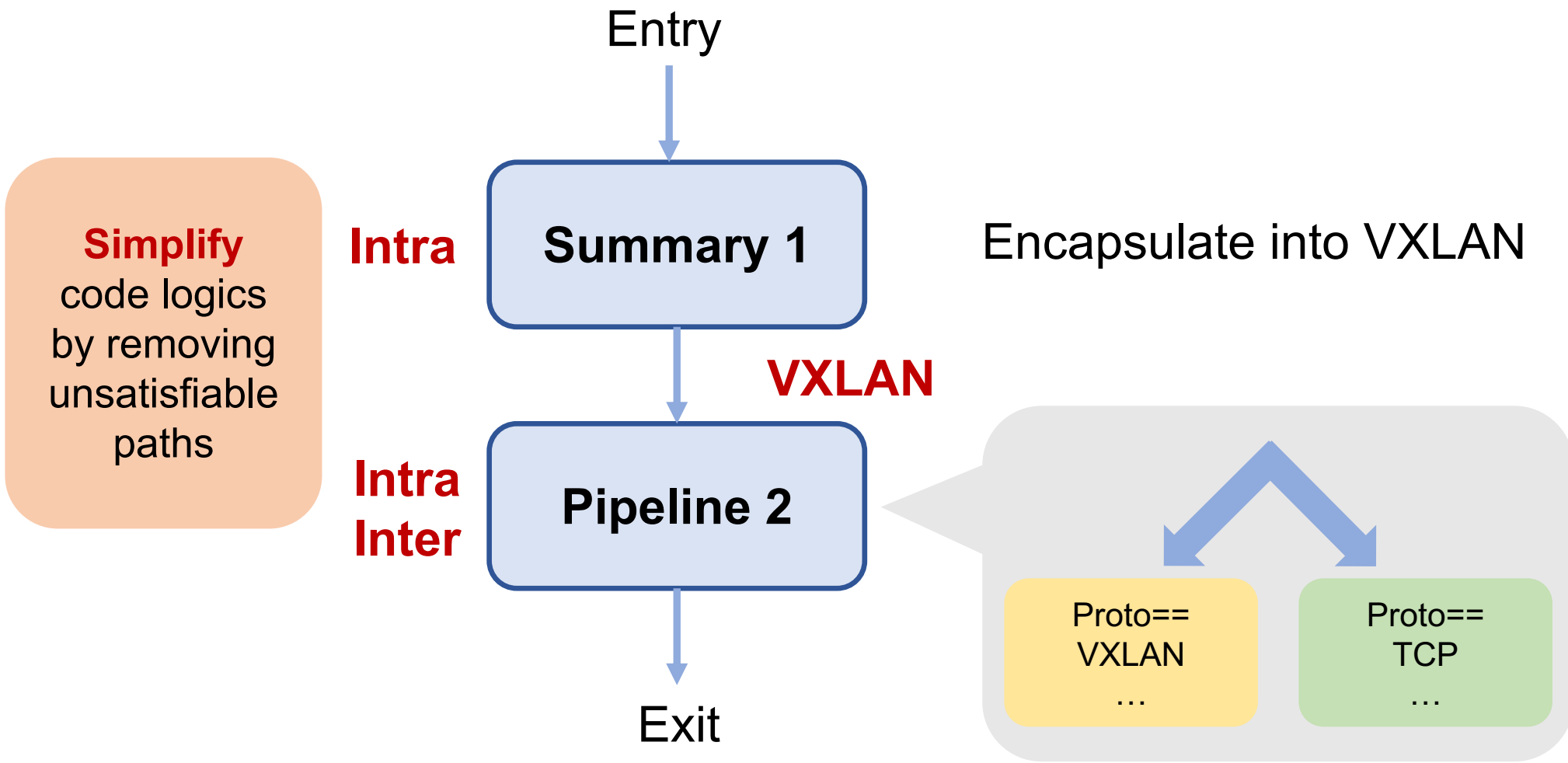
Keep valid path contexts

DFS

Symbolic execution



Pipeline simplification



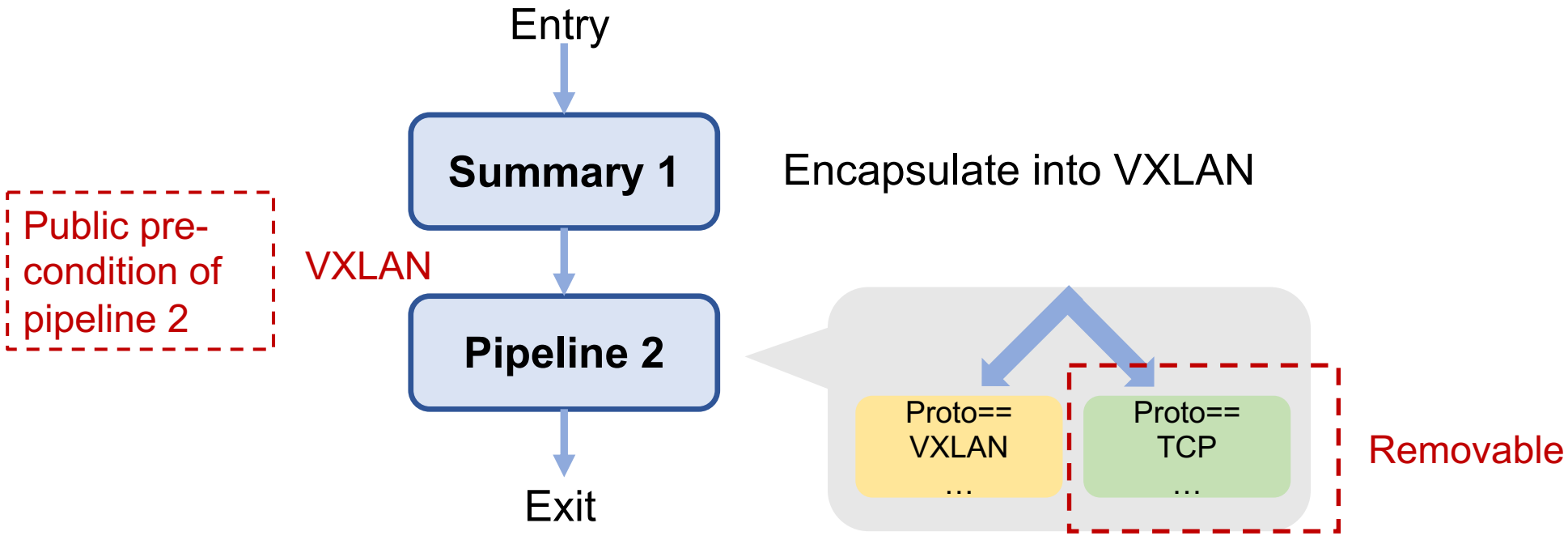
Inter-pipeline
public pre-condition filtering

Public pre-condition:

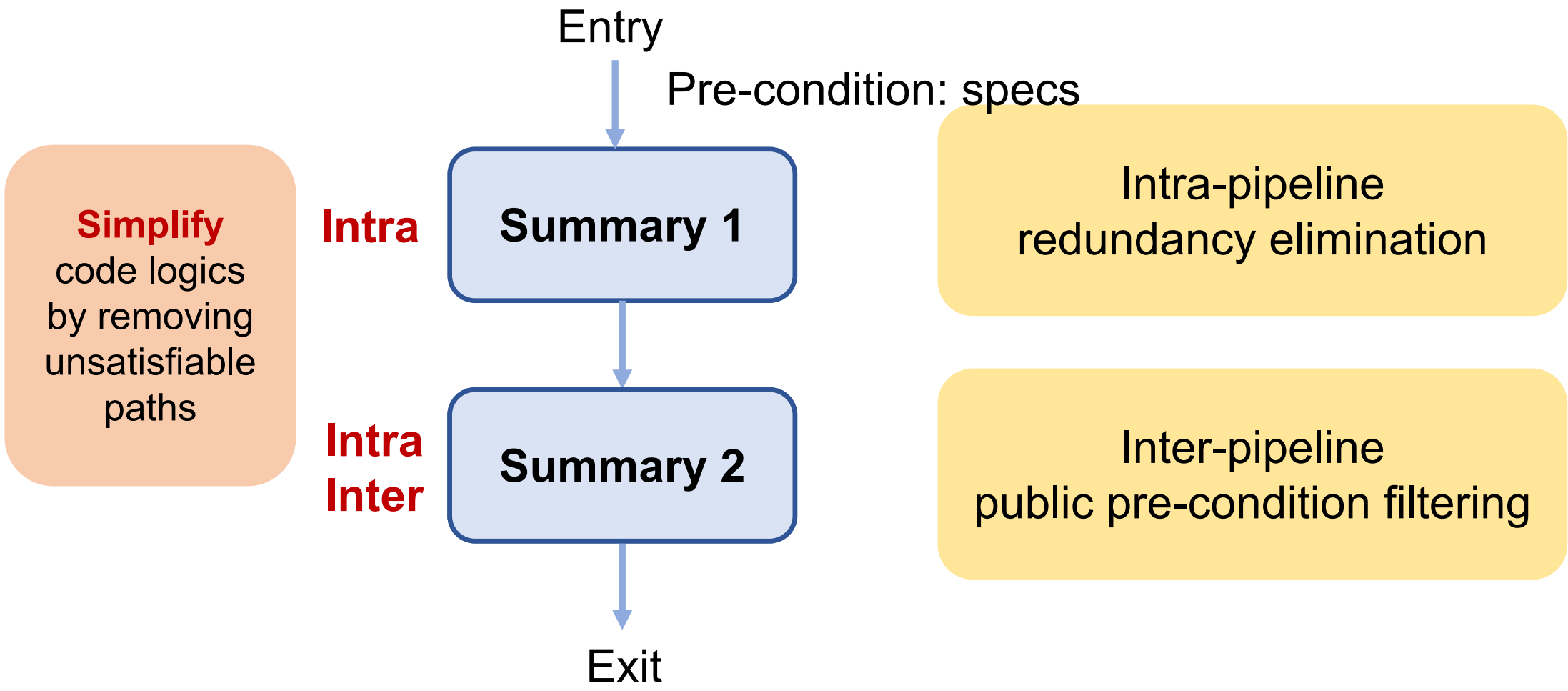
The common conditions at the beginning of target pipeline.

Methodology:

Find conditions of all paths from entry to target pipeline.



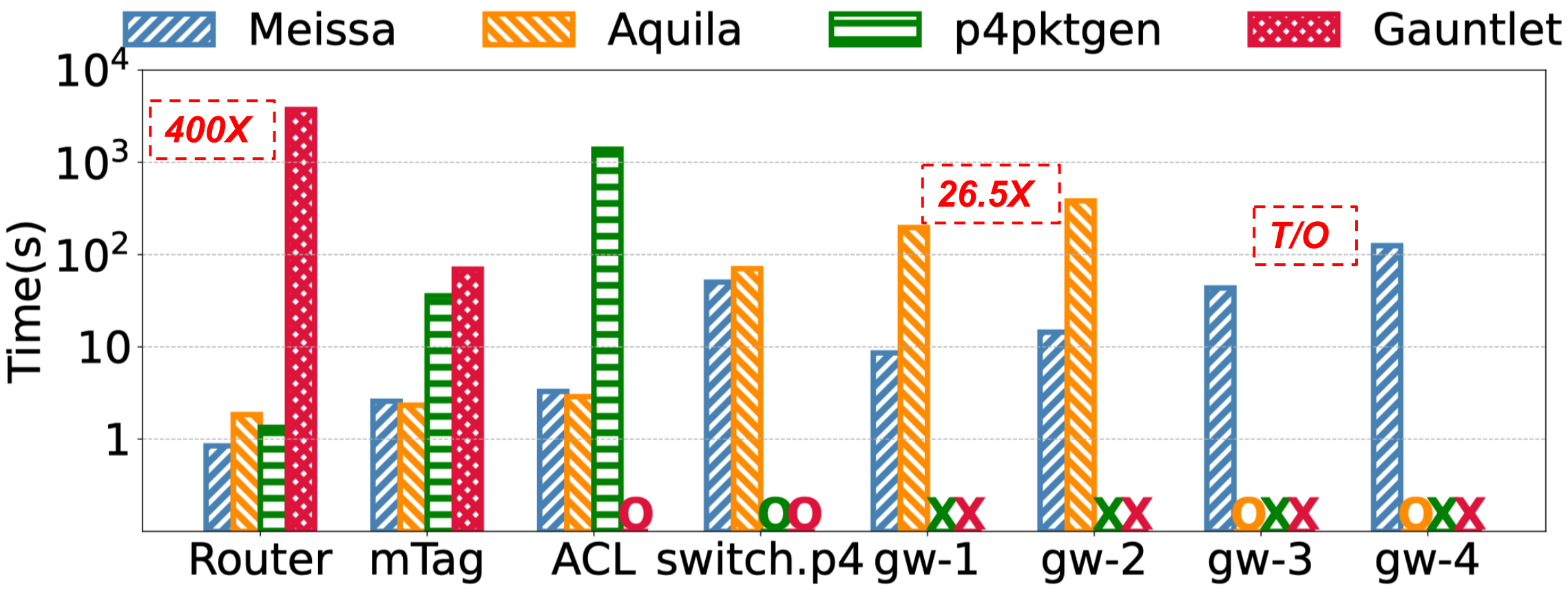
Pipeline simplification



Evaluation methodology

	Name	LOC	# of pipelines	# of switches
Open-sourced	Router	256	1	1
	mTag	227	1	1
	ACL	400	1	1
	switch.p4	7086	1	1
Industrial production	gateway-1	>1000	1	1
	gateway-2	>3000	2	1
	gateway-3	>10000	4	1
	gateway-4	>20000	8	2

Scalability



O for time-out, X for non-support

Bug finding ability

Type	Index	Bug	Meissa	p4pktgen	PTA	Gauntlet	Aquila
Code Bugs	1	Routing misconfiguration	✓	✗	✗	✗	✓
	2	Unrestricted ACL rules	✓	✗	✗	✗	✓
	3	Parser wrong logic	✓	✓	✓	✓	✓
	4	Ingress wrong logic	✓	✓	✓	✓	✓
	5	Wrong deparser emit	✓	✗	✓	✗	✓
	6	Checksum fail-to-update	✓	✗	✗	✗	✗
Non-code Bugs	7	p4c frontend bug 2147	✓	✓	✗	✓	✗
	8	p4c frontend bug 2343	✓	✓	✗	✓	✗
	9	bf-p4c backend bug 1	✓	✗	✗	✓	✗
	10	bf-p4c backend bug 3	✓	✗	✗	✓	✗
	11	bf-p4c backend bug 6	✓	✗	✗	✓	✗
	12	bf-p4c backend bug A (incorrect arithmetic comparison)	✓	✗	✗	✗	✗
	13	bf-p4c backend bug B (incorrect assignment)	✓	✗	✗	✗	✗
	14	bf-p4c backend bug C (setValid)	✓	✗	✗	✗	✗
	15	Misuse of optimization pragmas	✓	✗	✗	✗	✗
	16	Missing compilation flags	✓	✗	✗	✗	✗

Unknown bugs

Meissa is widely deployed

Since fall 2021, Meissa has been deployed in more than 200 P4 programmable gateways among 4 continents.

Finding real bugs

Non-code bug: Misuse of optimization pragmas

Misuse of pragmas

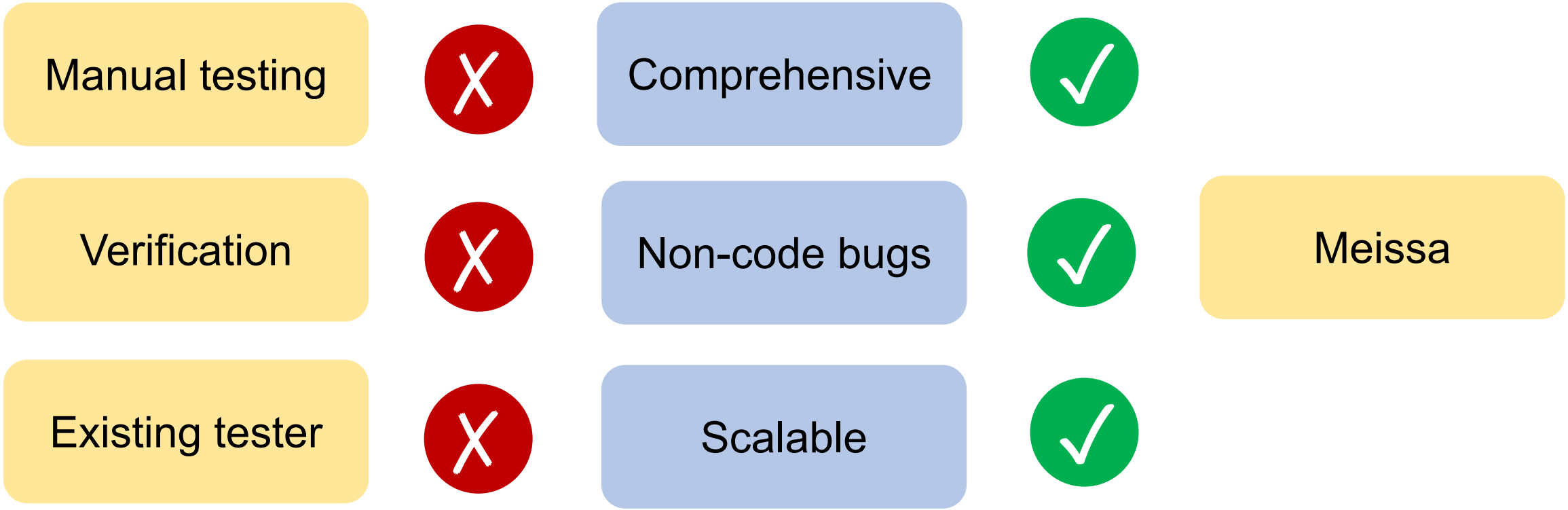
Two fields overlapped

Incorrect ACK of output packet

Non-code bug

Finding real bugs

Non-code bug: Misuse of optimization pragmas



Conclusion

Meissa is a scalable network testing system for programmable data planes.

Meissa leverages a domain specific code summary technique to guarantee full coverage and scalability.

Meissa is developed for programmable switches, but its principals also apply to other programmable data plane devices.

Thanks!

nq.zheng@pku.edu.cn