

# Attackboard: A Novel Dependency-Aware Traffic Generator for Exploring NoC Design Space

Yoshi Shih-Chieh Huang, Yu-Chi Chang, Tsung-Chan Tsai, Yuan-Ying Chang and Chung-Ta King  
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

## Motivation

Trace-driven simulation is simple and fast for exploring NoC design space. For the consideration of accuracy, traces with packet dependencies is necessary.

However, these trace logs can be very complicated and require large storage space!

The BIG problem is **size**, while the conventional trace compression mechanism is not a good solution for reducing the size of traces while maintaining accuracy.

## Key Question

**How to reduce size of traces while maintaining accuracy?**

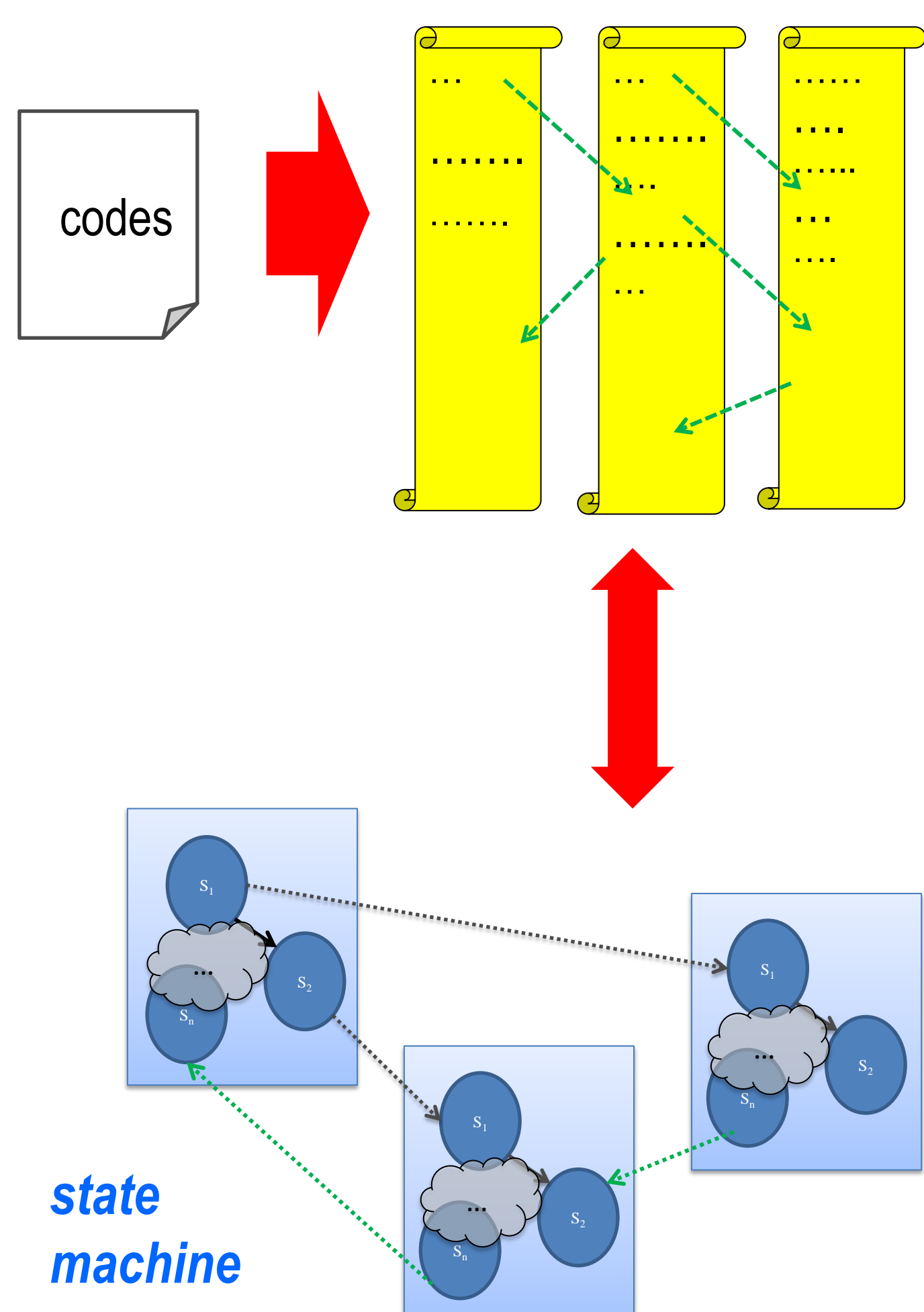
**Domain:** NoC design space exploration  
**Proposal:** A novel pattern-driven simulation mechanism

## Key Insights

- Each PE has its own BIG trace for NoC operations
- Each BIG trace is actually a log of the execution of the corresponding **State Machine**

10KB codes may result in more than 1GB traces!

1GB trace logs



## Rebuild The State Machine

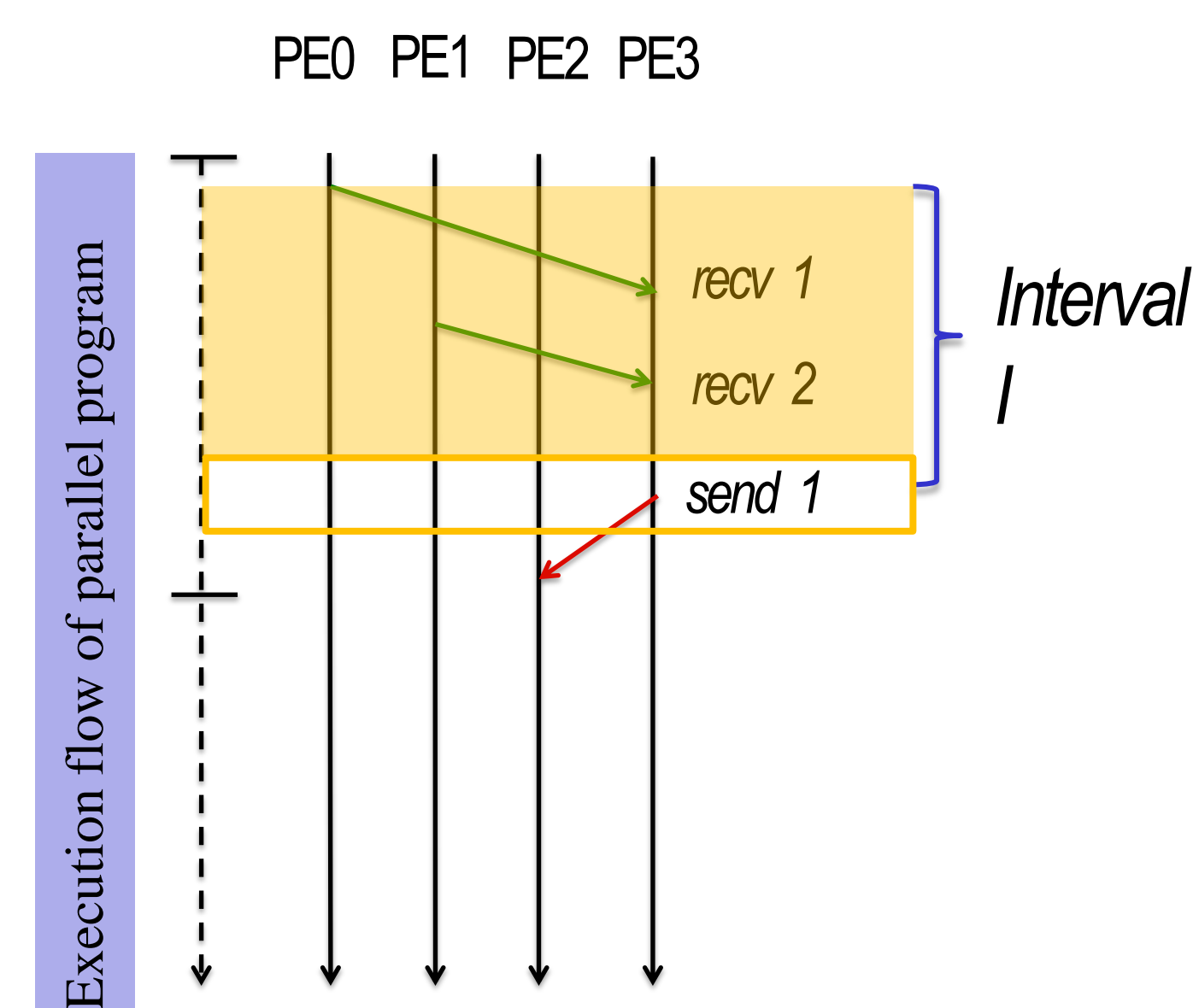
**Attackboard uses small tables to represent the state machines**

### How to do?

- State transitions are triggered by arrivals of packets
  - Leverage packet dependency info. in traces
  - Focus on patterns of received packets

### Key idea #1

- Time-driven to pattern-driven
  - Use packet arrival patterns to replace the time sequencing
  - Inject packets by the sequencing of states



Packets Dependencies	Injection Info.
1 1 0 0	(2, flit counts)

## Table Minimization

### Key idea #2

- Merge entries with the same patterns
  - Fold the repetitive patterns: merge the entries with the same packet dependencies
  - Merge duplicated entries

Packets Dependencies	Injection Info.
1 1 0 0	(2, flit counts)

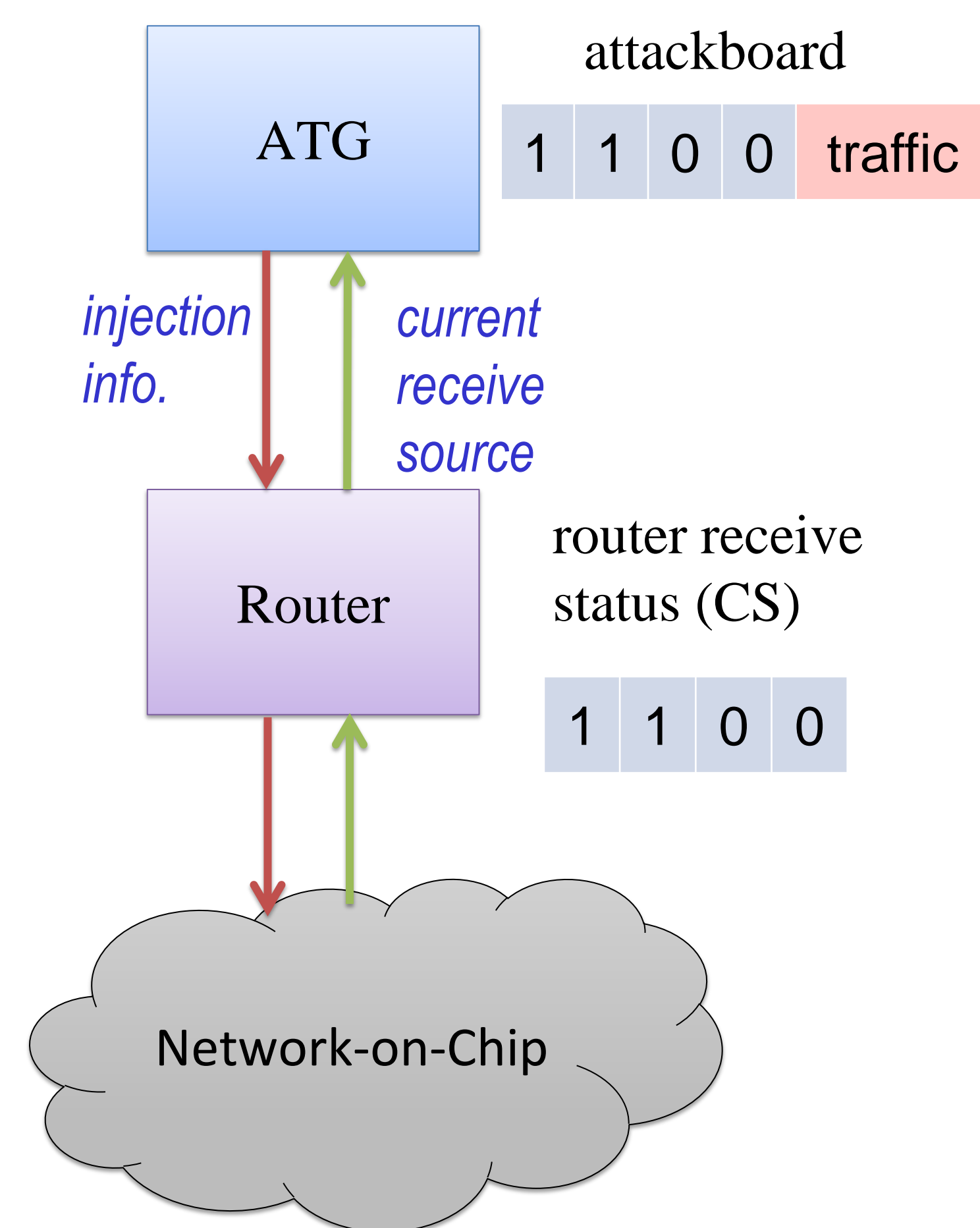
Packets Dependencies	Injection Info.
1 1 0 0	(0, flit counts)

Packets Dependencies	Injection Info.
1 1 0 0	(0, flit counts)

Packets Dependencies	Injection Info.
1 1 0 0	(2, flit counts)
1 1 0 0	(0, flit counts)

## Overview of Attackboard

Attackboard Traffic Generator (ATG)



## Attackboard Simulation

START

Match the CS with attackboard

### Key idea #3:

- Match patterns with multiple bloom filters
  - Deny for sure, allow for high probability
  - 0 in the packet dependencies will be considered as "don't care"
  - E.g. pattern "1110" will match "1100" when "1110" does not exist in attackboard

Select the traffic

- Select the traffic in injection info.
  - The matched result of attackboard and CS is a set of send events
  - 3 ways to select the traffic
    - Averaged
    - Circular Queue
    - Probability

Packets injecting, forwarding and receiving

- Simulate the packets injection and transmission
  - Router changes its CS at the arrival of the packets

Is traffic generation interval I' expired?

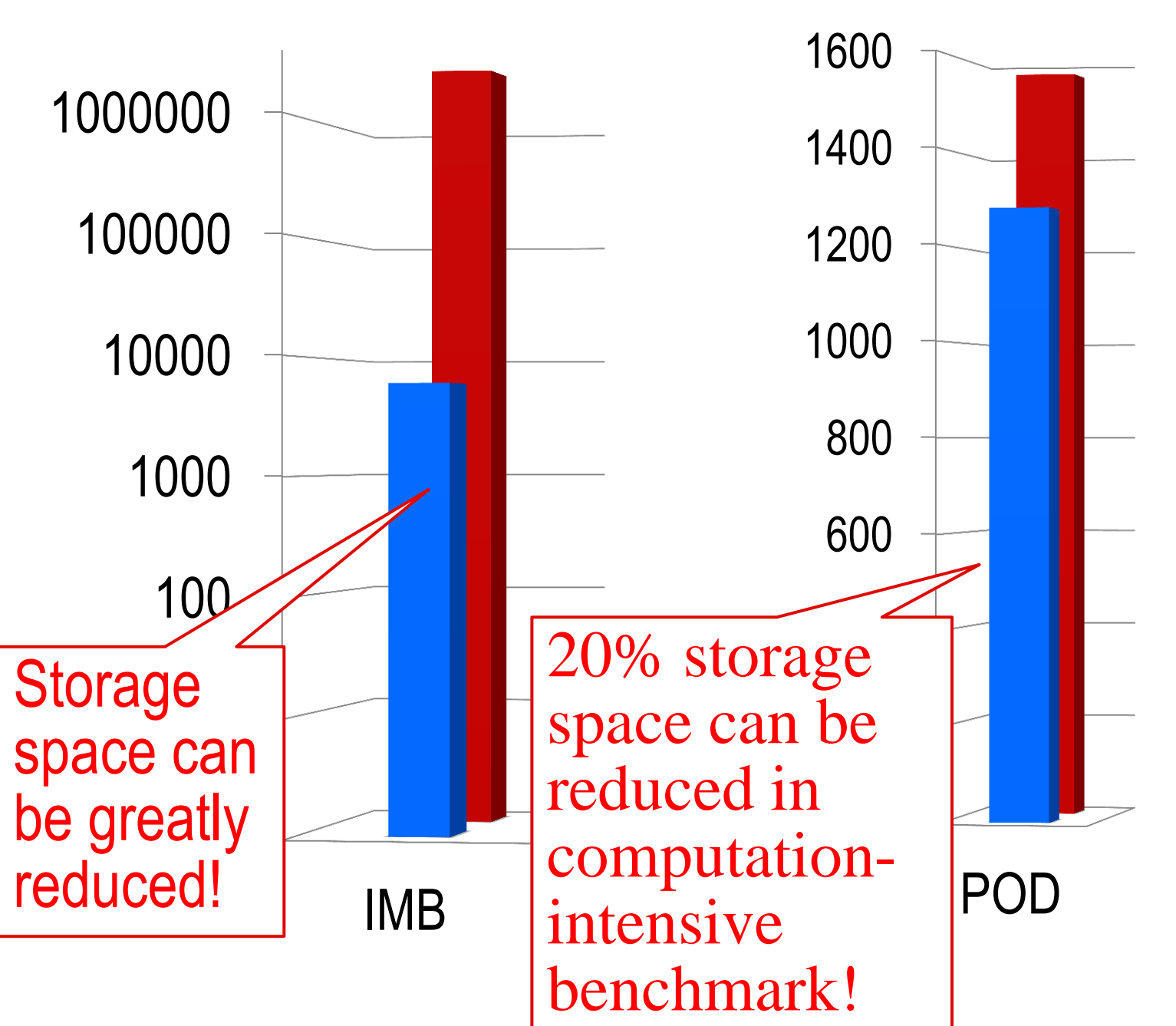
- I' has not expired
  - Keep simulating packets transmission
- I' has expired
  - Start a new simulation flow

## Evaluation Setup

### Emulate ATG on Tiler TILE64

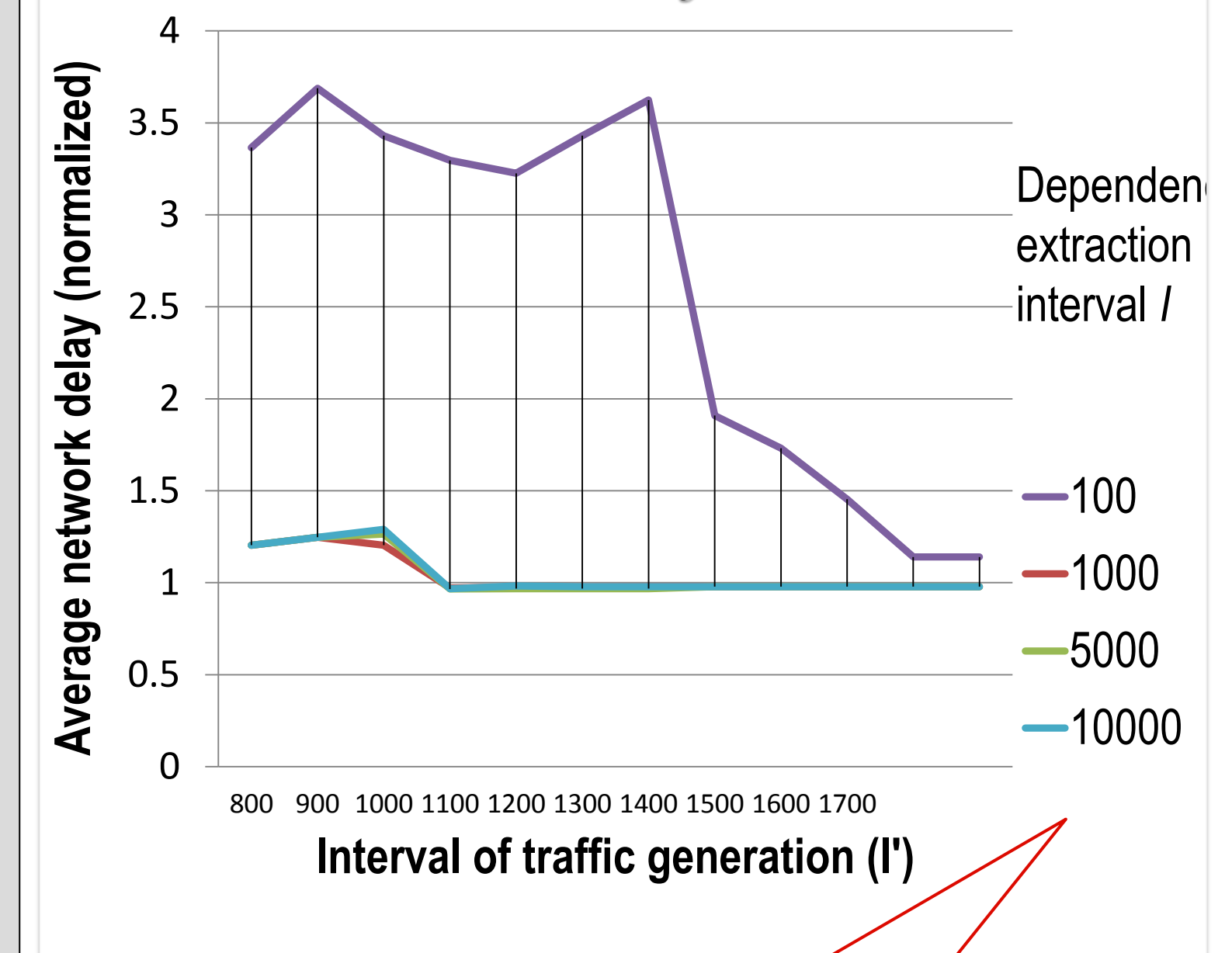
Native processor element	Tiler TILE64
Native processor frequency	700 Mhz
Simulated topology	4x4 mesh network
Routing algorithm	Dimension-order
Bandwidth	1 flit/cycle per port
Benchmark	Intel MPI Benchmarks Parallel Object Detection

## Storage Space Overhead

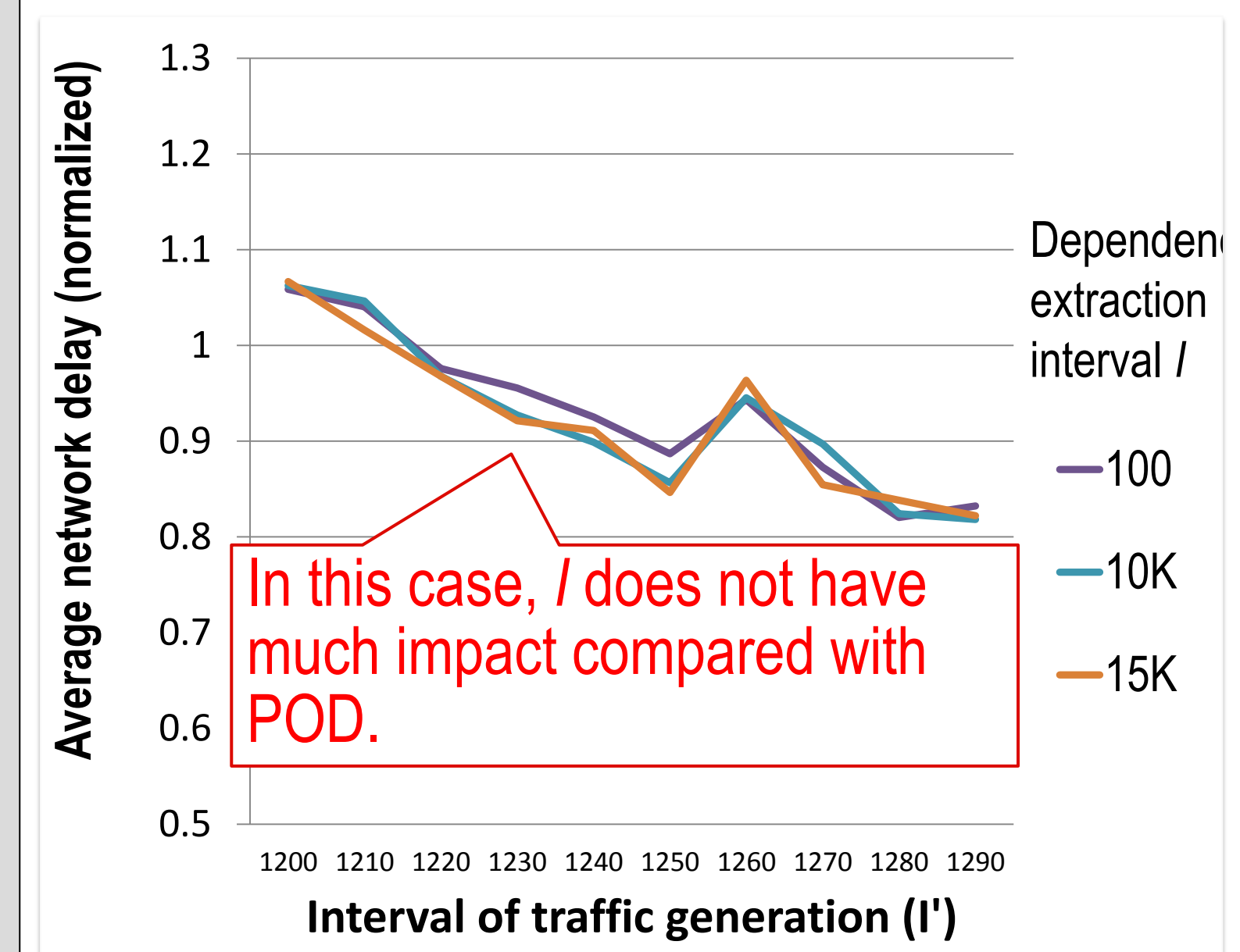


## Accuracy Evaluation

### Parallel Object Detection



### IMB Broadcast



## More Information

More about this paper!



More about PADS Lab @NTHU CS

