Bertacco Lab
Designing correct heterogeneous systems

Design Space Exploration of Heterogeneous System Interconnect

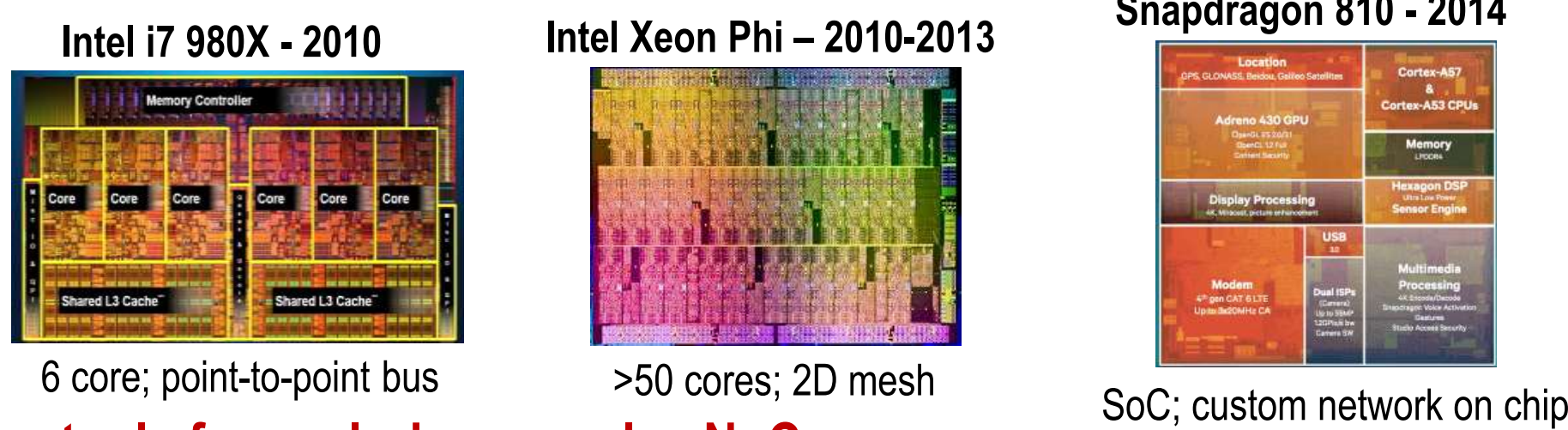## Dong-Hyeon Park, Vaibhav Gogte, Ritesh Parikh and Valeria Bertacco

1. Motivation

Architectures are moving toward Heterogeneous Computing
- Future NoCs need to support wide range of complex IPs
- Interconnect designs need to be robust and flexible

Intel i7 980X - 2010
6 core; point-to-point bus

Intel Xeon Phi – 2010-2013
>50 cores; 2D mesh

Snapdragon 810 - 2014
SoC; custom network on chip

Two tools for exploring complex NoCs

PacketGenie:
Traffic Generator for interconnects of Heterogeneous Systems

NoCVision:
Traffic Visualizer for Network-on-Chip Systems

2. PacketGenie

PacketGenie Purpose:
- Allow **quick design-space exploration** of heterogeneous NoC interconnect
- Extract application behavior and generate traffic based on system model

PacketGenie Goals:
- **Quick, light-weight simulation** of NoC traffic behavior
- **Highly flexible and configurable** to accommodate heterogeneous architectures
- **Easy to integrate** with existing network simulators

Heterogeneous System
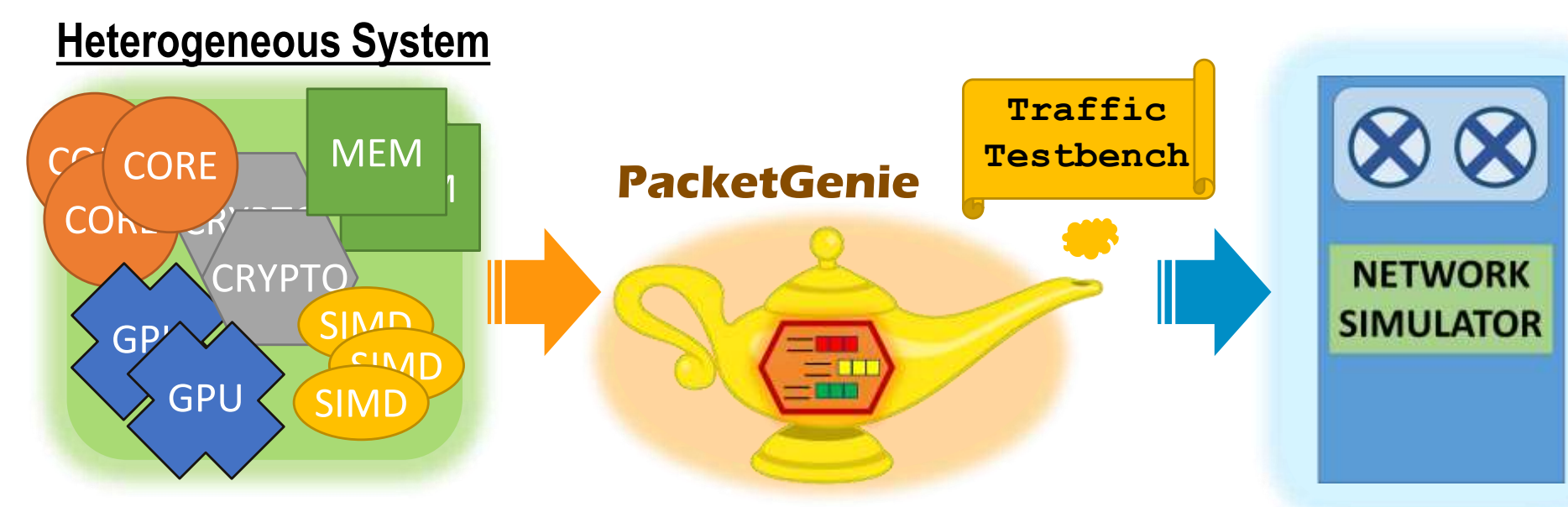
3. NoCVision
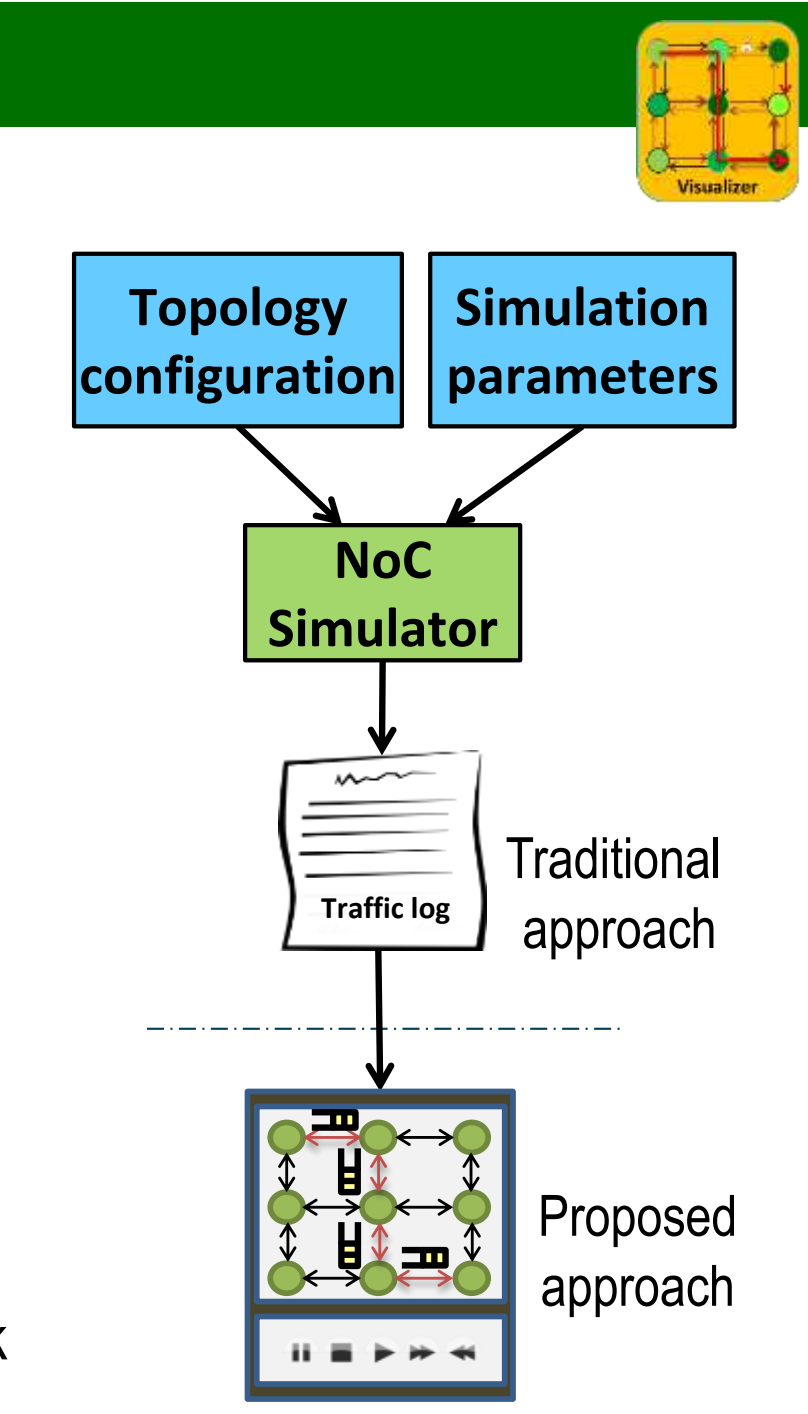
Debugging complex NoCs: Issues
- Analyze millions of cycles
- Debug large logs

Proposed debug approach
- Graphical representation of traffic flow
- Extraction of relevant traffic details e.g. network congestion, bottlenecks etc.
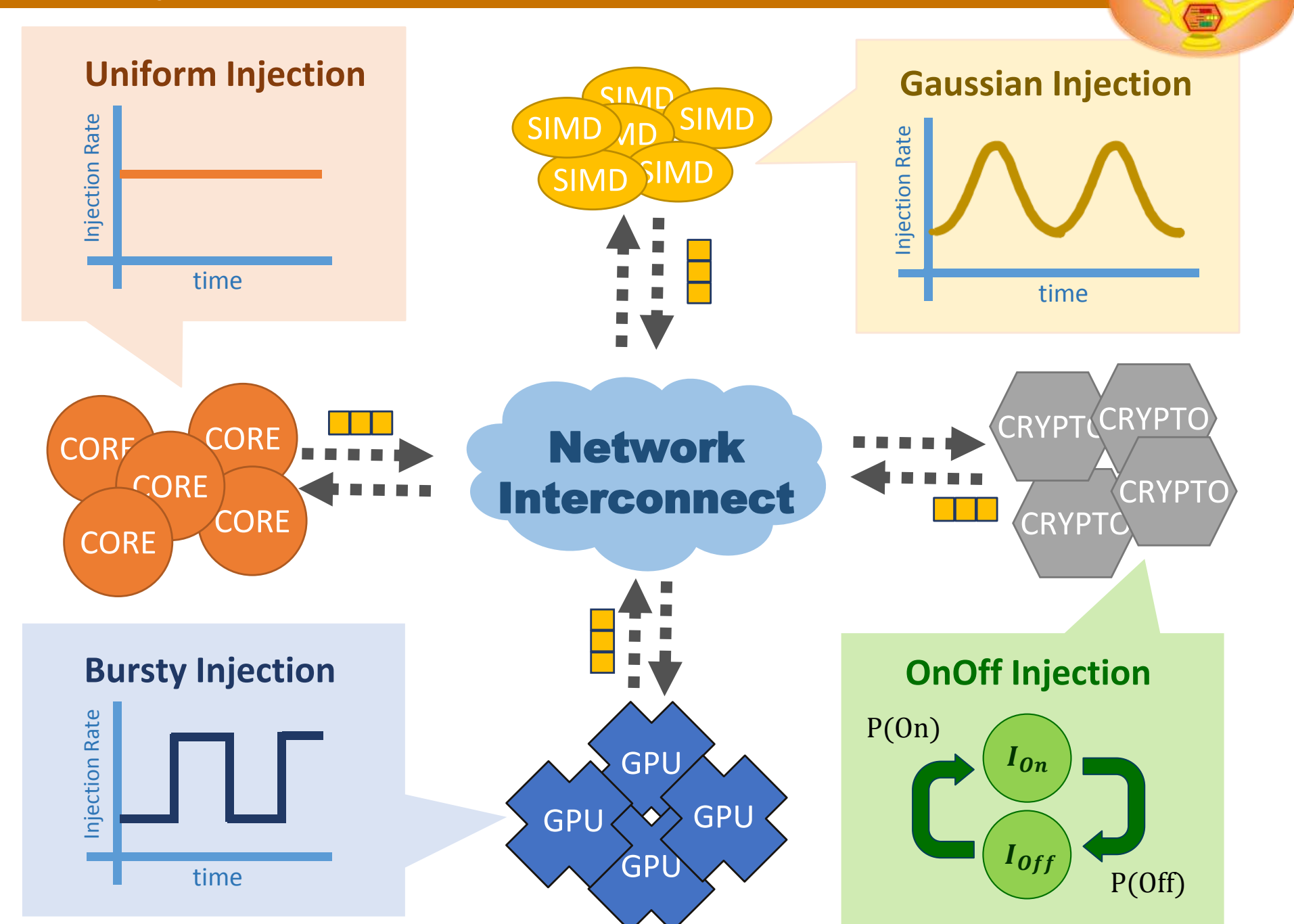
Implementation
- Capturing interval-based and event-based packet flow
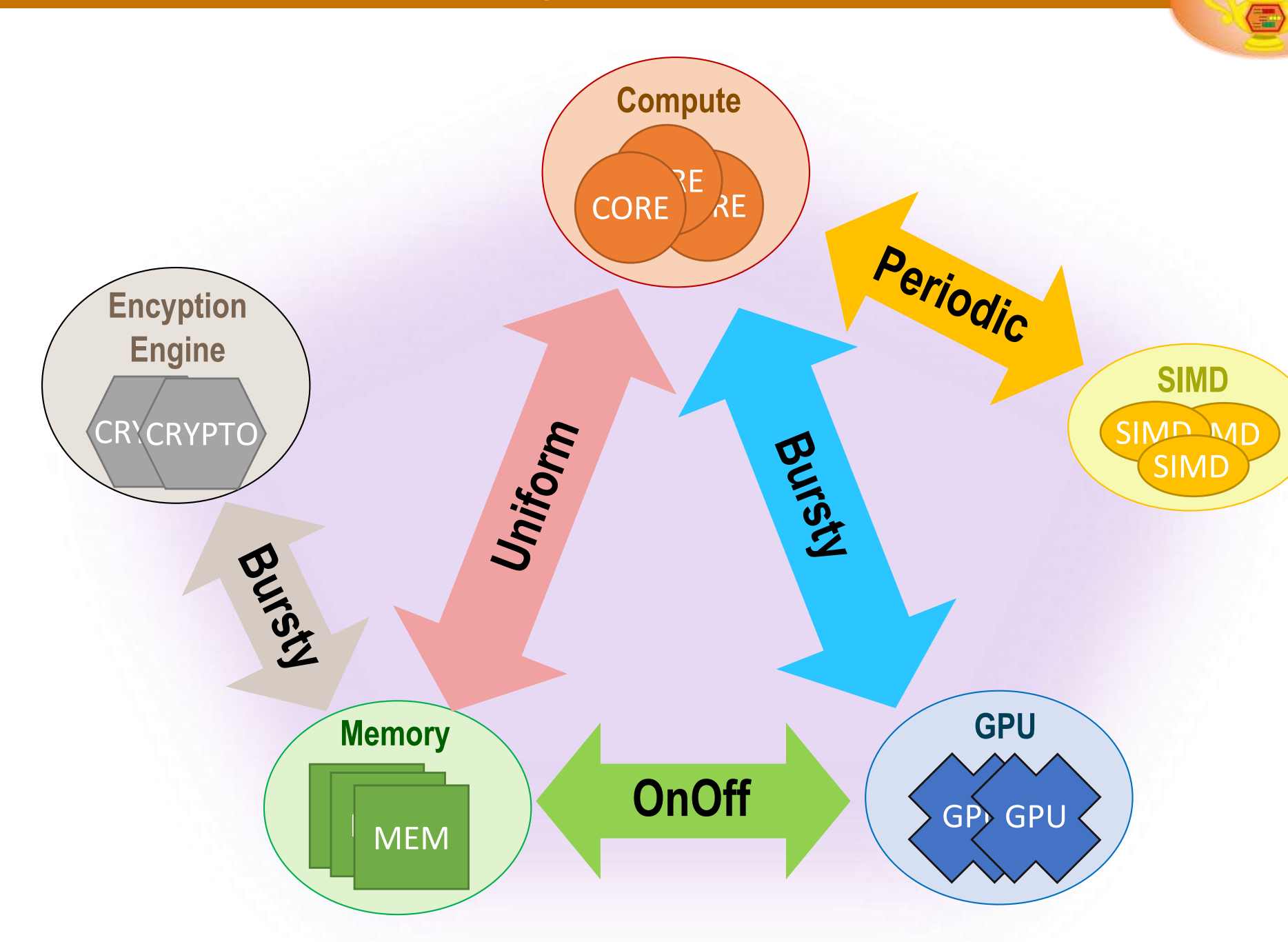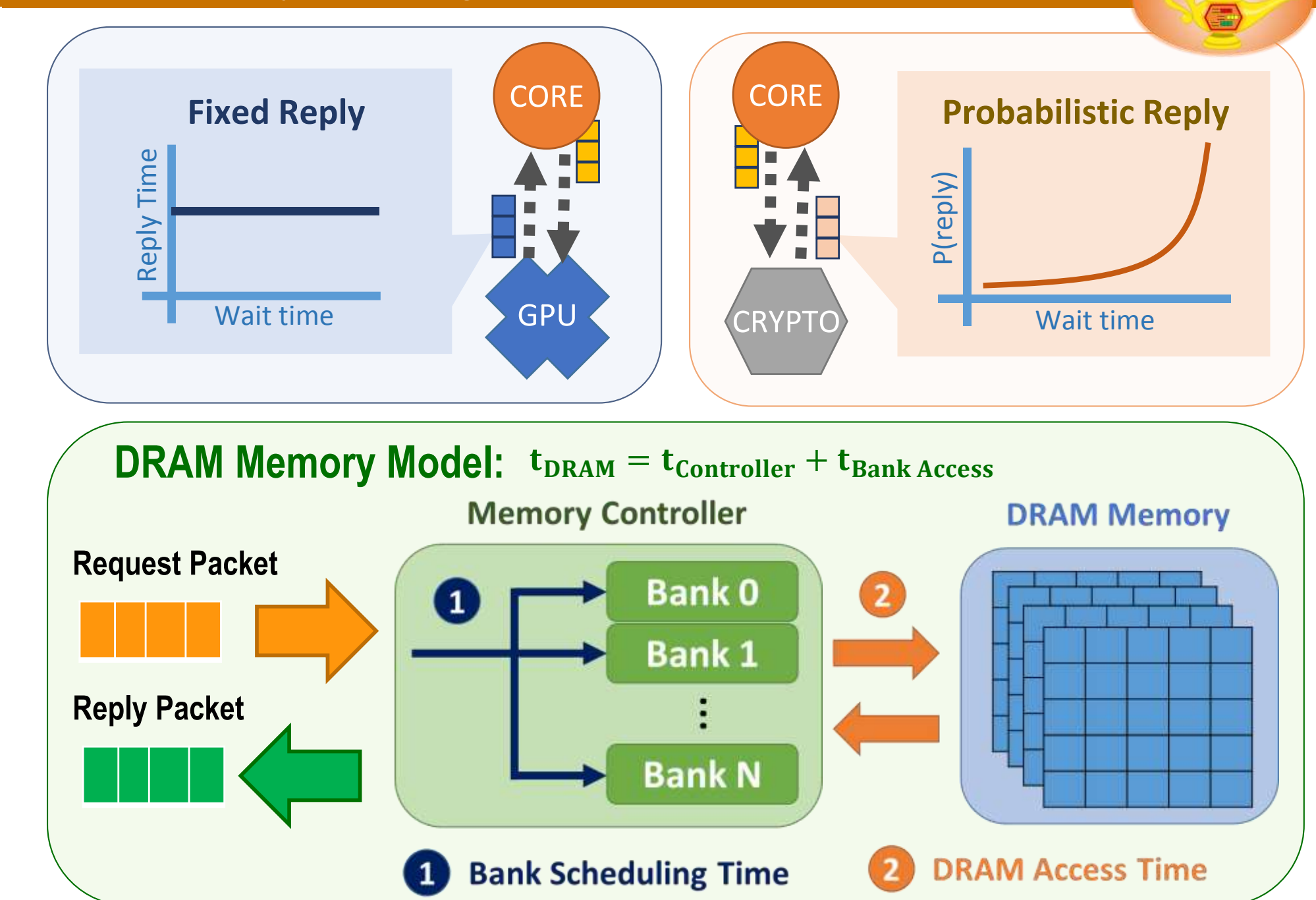- Visualization of a network topology: router, link or VC level information

Traditional approach

Proposed approach

PacketGenie

4. Injection Models

Uniform Injection

Gaussian Injection

Bursty Injection

OnOff Injection

Network Interconnect

5. Example Configuration

Compute
Encryption Engine
Periodic
SIMD
Uniform
Bursty
Memory
Bursty
OnOff
GPU

6. Reply Timing Model

Fixed Reply

Probabilistic Reply

DRAM Memory Model: $t_{DRAM} = t_{Controller} + t_{Bank\ Access}$

Memory Controller

Request Packet

Reply Packet

Bank 0
Bank 1
⋮
Bank N

DRAM Memory

① Bank Scheduling Time   ② DRAM Access Time

NoCVision

7. NoCVision Flow

Network topology configuration

time intervals
events

Mode of operation - Interval mode or event mode

Traffic data associated with router, link or VC of NoC

Link congestion

Router utilization

Packet Traversal

Parameter Threshold
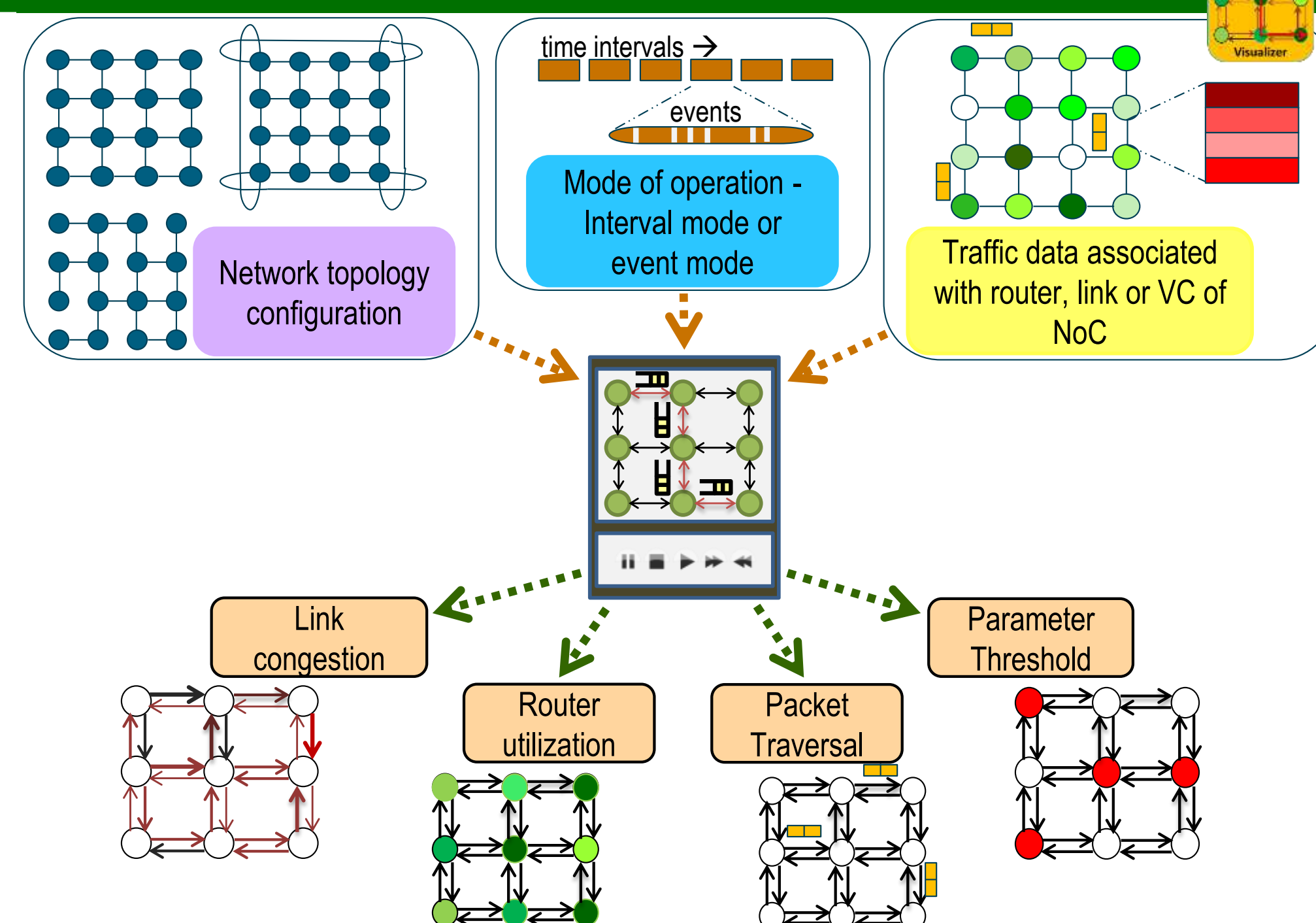
8. NoCVision – Modes of Operation

Type 1: Interval Mode
- Interval-based packet flow
- Link, router and VC utilization across time windows
- Color intensity represents parameters e.g. congestion, utilization etc.

Interval 1   Interval 2

Heavily utilized routers

Link gets congested in the next interval
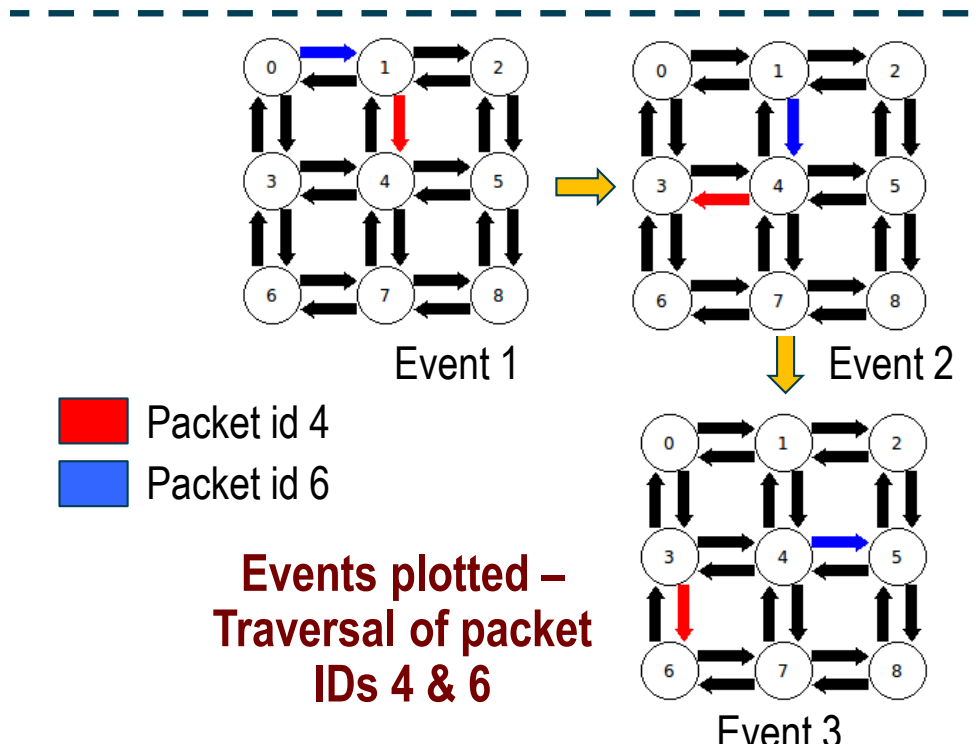
Type 2: Event Mode
- Determine region-of-interest and log data for specific events
- Parse through the data across events

Event 1   Event 2
Event 3

Packet id 4
Packet id 6

Events plotted – Traversal of packet IDs 4 & 6

9. Case-study – High Radix vs. Low Radix

Application-adaptive topology reconfiguration:
- Employs routers with few ports as in **low-radix** topologies and many links as in **high-radix** routers
- Reconfigures network to enable low-latency path between **heavily communicating src-dest pairs**

NoCVision was used to analyze the routing in various topologies: low-radix, high-radix and hybrid

Longer path due to routing restrictions

Heavily communicating pairs (color indicates the pairs)

Low-radix: 2D mesh Avg Path Length = 5

High Radix

Hybrid router

Hybrid: Adaptive 3D torus Avg Hop Count = **3.5**

High-radix: 3D torus Avg Path Length = **3**