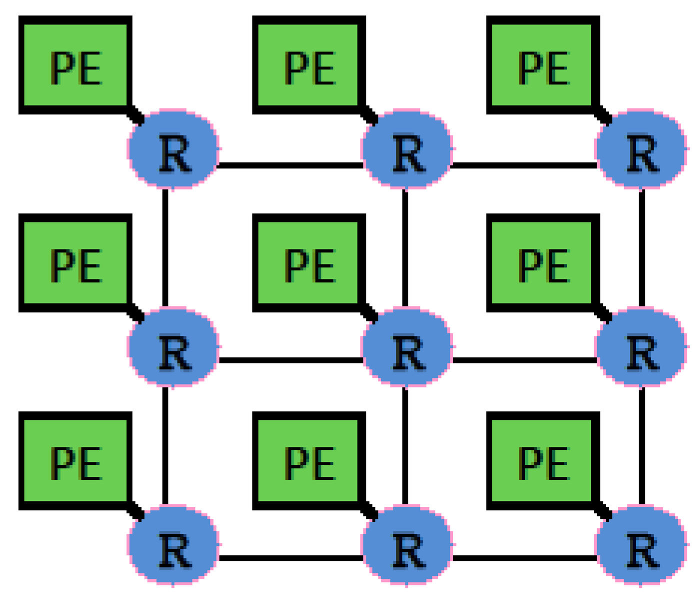
HAT: Heterogeneous Adaptive Throttling for On-Chip Networks

Kevin Kai-Wei Chang, Rachata Ausavarungnirun, Chris Fallin, Onur Mutlu (Carnegie Mellon University)

Background and Problem

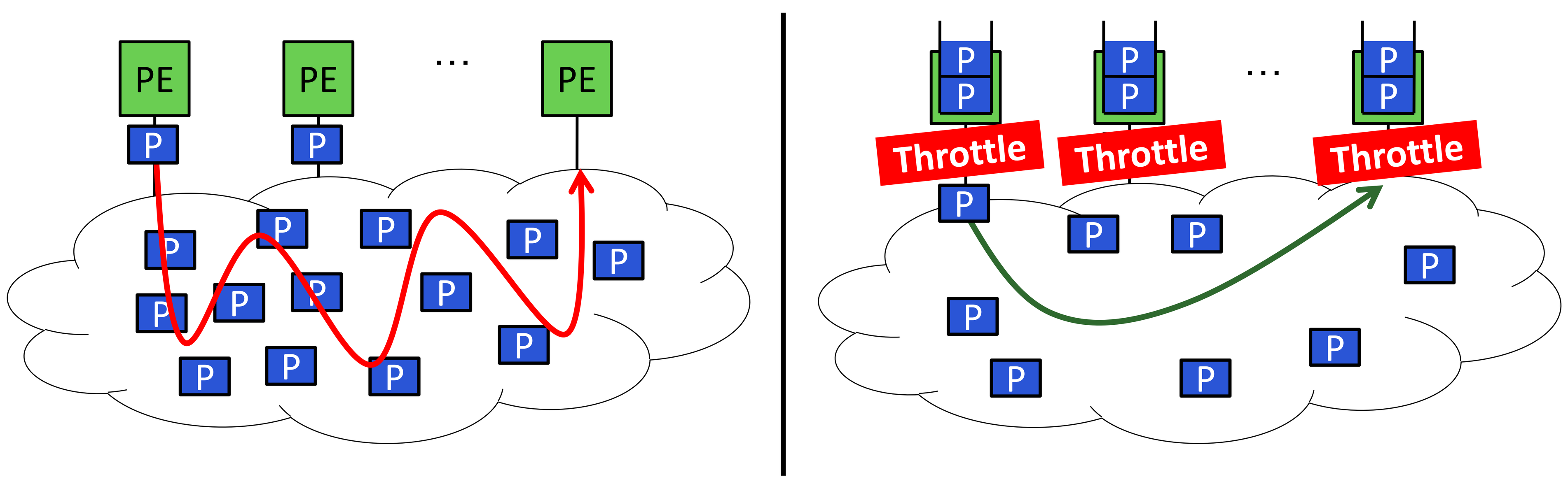


R Router
PE Processing Element (Cores, L2 Banks, Memory Controllers, etc)

- Network has limited shared resources (buffers and links) due to on-chip design constraints (power, die size, wiring)
- Problem: Packets contend in on-chip networks (NoCs), causing network congestion, thus reducing system performance

Motivation

- Goal: Improve system performance in a highly congested network
- Observation: Reducing **network load** (number of packets in the network) decreases network congestion, hence improves system performance
- Approach: Source throttling (temporarily delaying new traffic injection) to reduce network load



- Throttling makes some packets wait longer to inject
- Average network throughput increases, hence higher performance

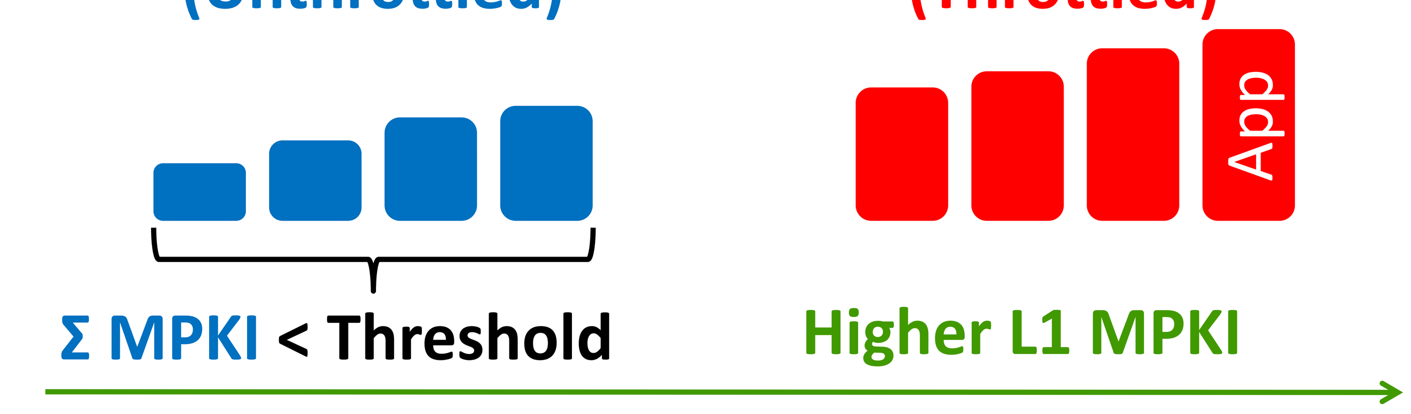
Key Questions:

1. Which applications to throttle?
2. How much to throttle?

Heterogeneous Adaptive Throttling

1 Application-Aware Throttling:

- **Key Observation**: Throttling network-intensive applications leads to higher system performance
  → Reduces network congestion significantly
  → Benefits both intensive and non-intensive applications, but non-intensive applications benefit more because they are more sensitive to network latency
- **Key Idea**: Throttle network-intensive applications that interfere with network-non-intensive applications
- **Mechanism**:
  1) Measure applications' network intensity: Use L1 MPKI (misses per thousand instructions)
  2) Throttle network-intensive applications: Select unthrottled applications so that their total network intensity is less than the total network capacity



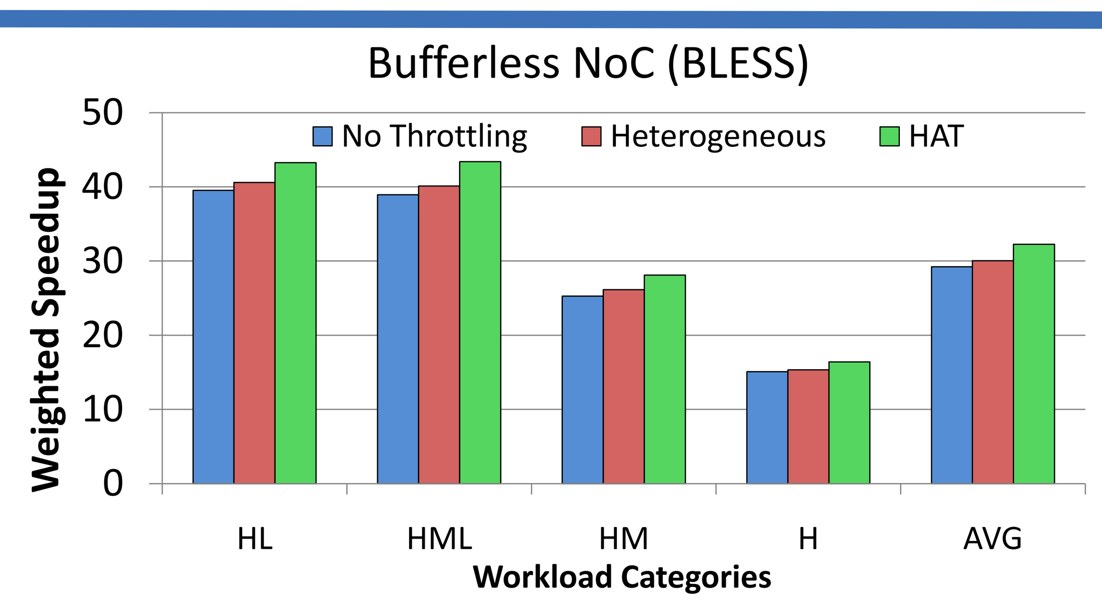
2 Network-load-aware throttling rate adjustment:

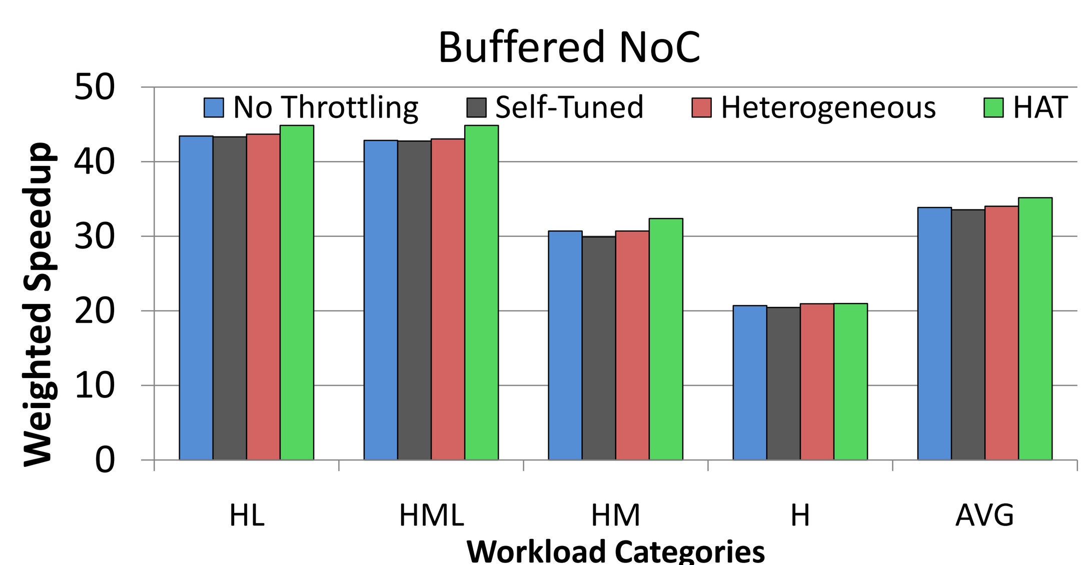
- **Key Observations**:
  1) There is no single throttling rate that works well for every application workload or program phase
  2) Network runs best at or below a certain network load, which is an accurate indicator of congestion
- **Key Idea:** Dynamically adjust throttling rate to adapt to different workloads and program phases
- **Mechanism:**
  1) Measure network load (fraction of occupied buffers/links)
  2) Dynamically adjust throttling rate to make the load stay close to the target network load

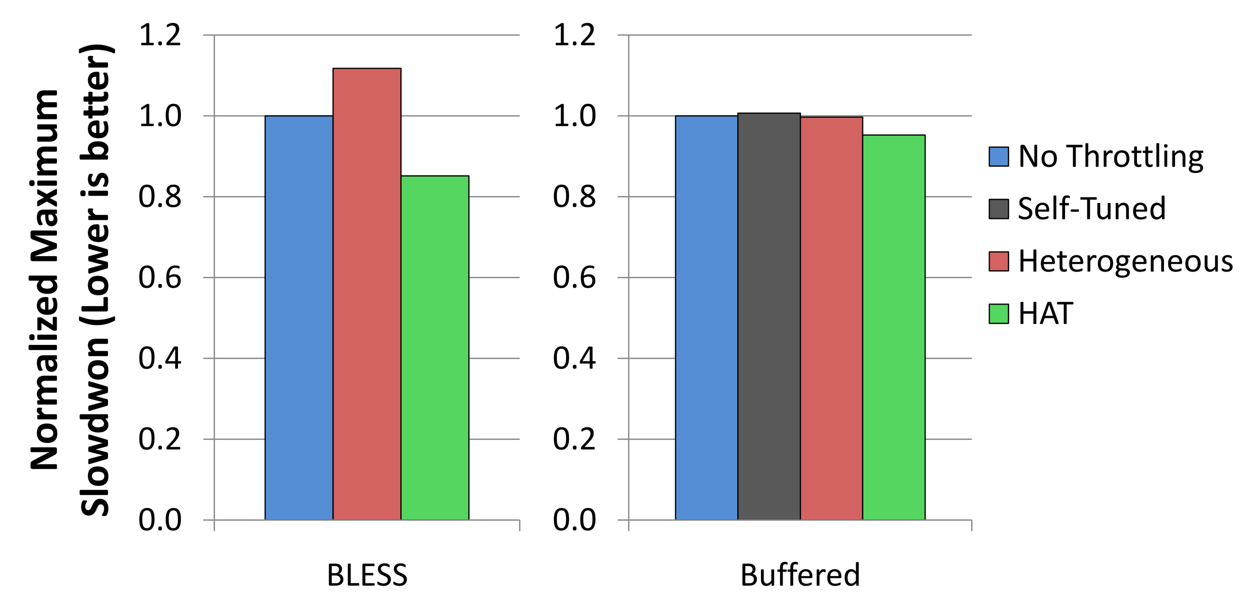
Results

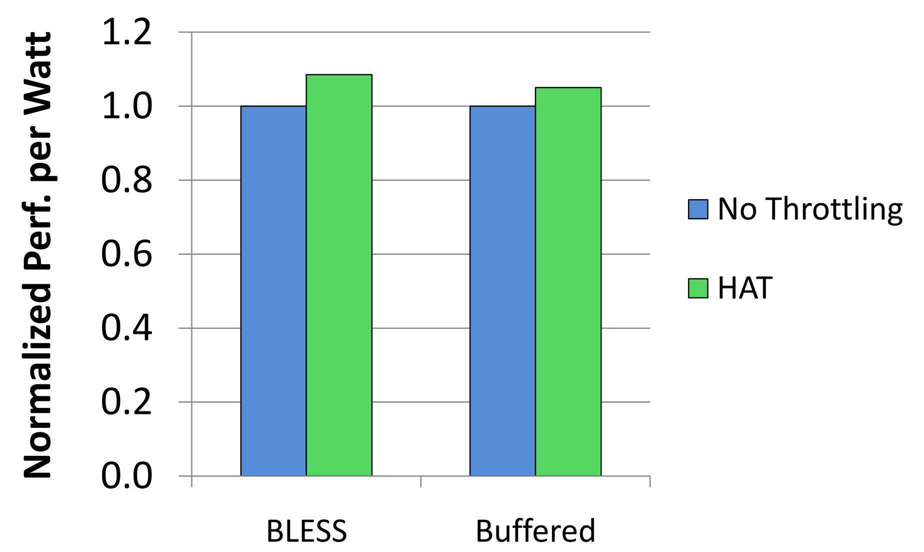
Methodology

- 64 OoO CPU cores with a 2D mesh
- 64KB L1, perfect L2 (always hits to stress NoC)
- Router Designs:
  1) Virtual-channel buffered router
  2) Bufferless deflection router: *BLESS*
- Workloads: Cloud-computing-like multiprogrammed combinations of CPU and memory intensive applications









Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University | Georgia Tech | intel | Princeton University | UC Berkeley