

---

# Low-power interconnection networks

---

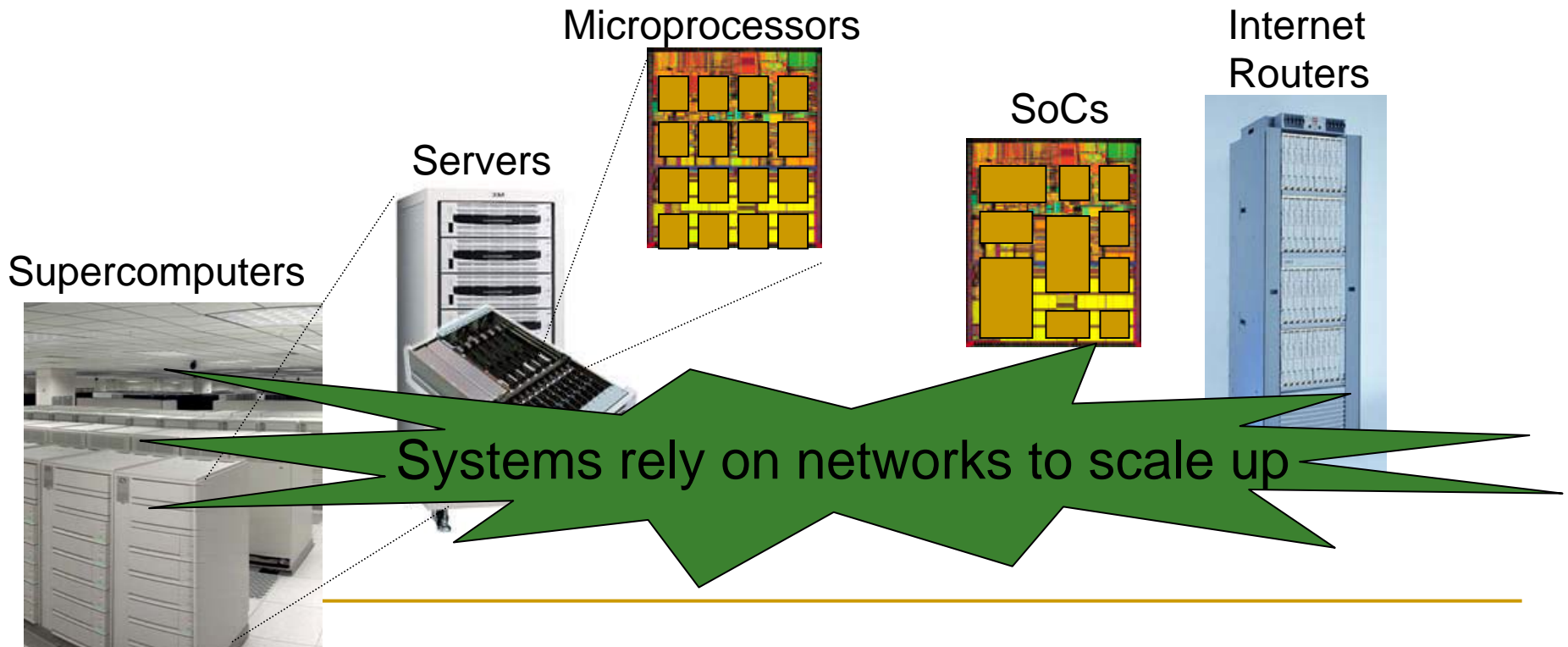
Li-Shiuan Peh  
Assistant Professor  
Department of Electrical Engineering  
Princeton University

# Why interconnection networks will be of increasing importance...

- Systems becoming increasingly parallel...

## General-Purpose Systems

## Application-Specific Systems



# Why we need low-power interconnection networks...

- 1. Systems are power-constrained due to cooling, power delivery & battery limits.*
- 2. Networks consume significant power.*

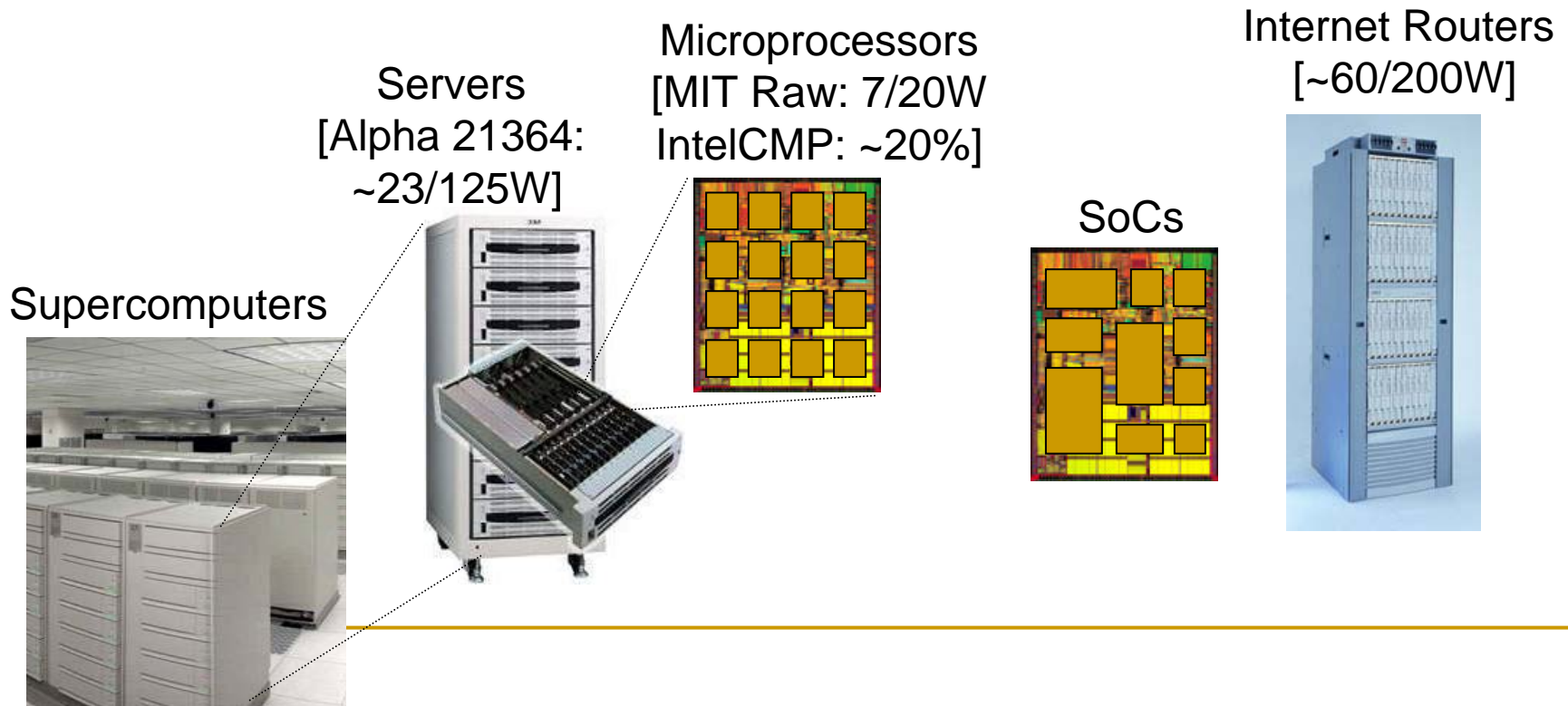
Supercomputers

Servers  
[Alpha 21364:  
~23/125W]

Microprocessors  
[MIT Raw: 7/20W  
IntelCMP: ~20%]

SoCs

Internet Routers  
[~60/200W]



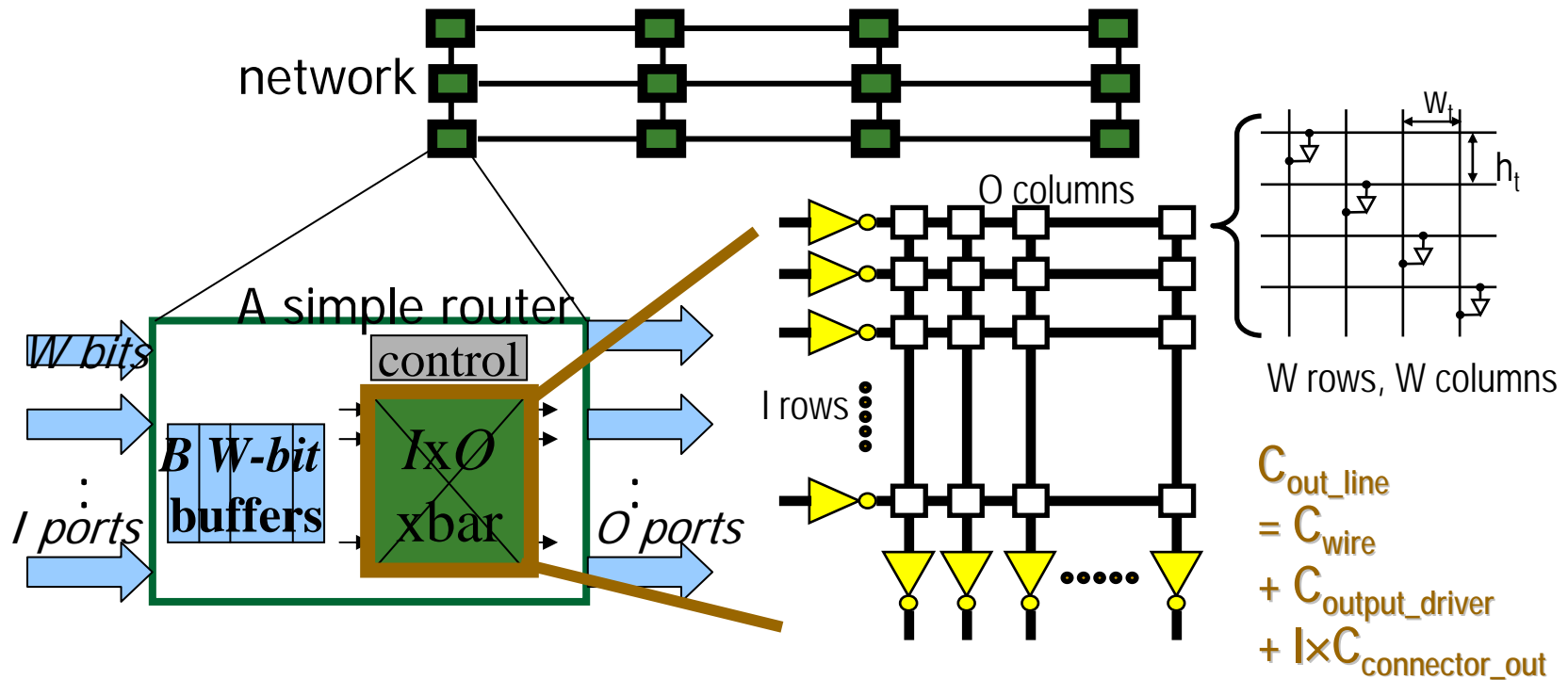
The image is a collage of five distinct visual elements. On the left, a photograph shows a large room filled with rows of white supercomputer cabinets. In the center, a server rack is shown with one of its drive trays pulled out. To the right of the server rack is a 4x4 grid of 16 small, yellow, square microprocessors. Further right is a single, larger yellow square representing a System-on-Chip (SoC). On the far right is a tall, blue internet router cabinet with many ports on its front panel. Dotted lines connect the labels to their respective images.

---

# Outline

- **Brief survey of group's research**
  - Power-driven network design tools
    - ORION: Architectural network power models
    - LUNA: High-level network power analysis
  - Power-aware networks
    - Dynamic voltage scalable networks
- Thermal modeling and management of on-chip networks
- Next: Network-driven computing

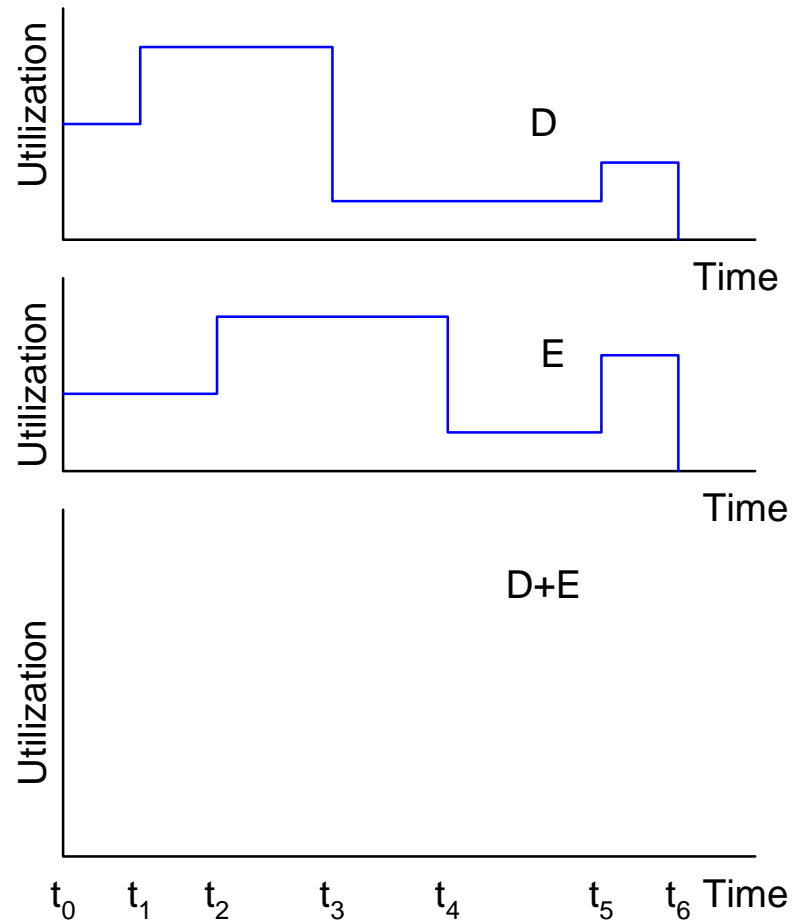
# ORION: Architectural network power models



- Validations: Alpha 21364, IBM InfiniBand switch (close to designers' estimates); MIT Raw (3-11% error)
- Status:
  - Used for network power estimation in CMPs, MPSoCs, network processors in academia and industry
  - 100+ downloads (as of May 2005)

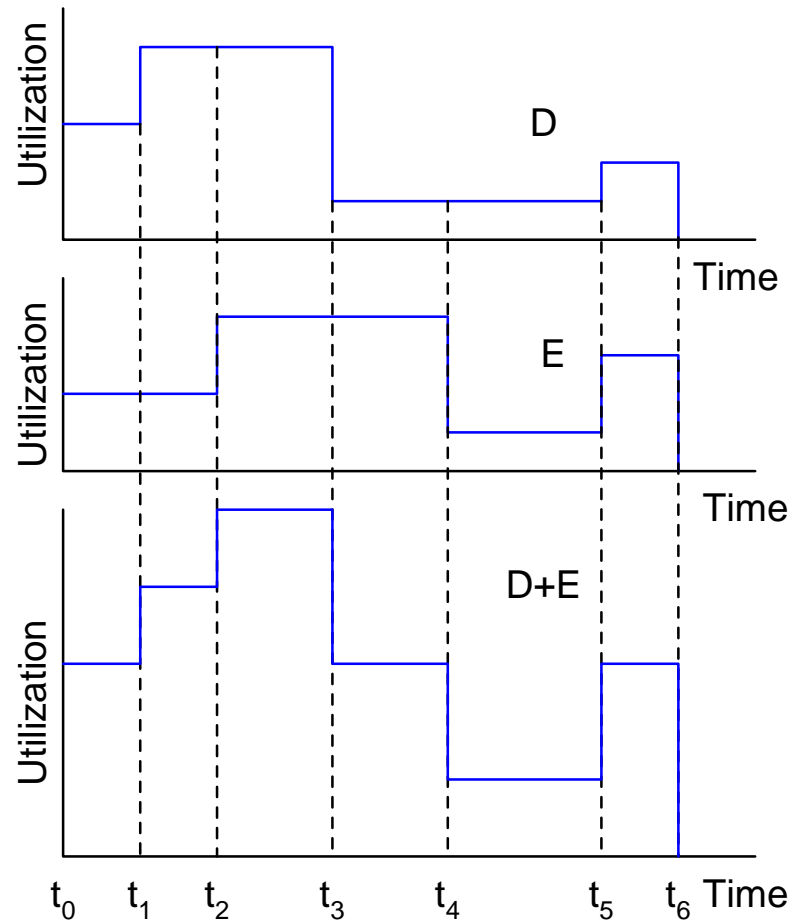
# LUNA: High-level network power analysis

- Link utilization as proxy for node power
  - Input: Message flows between nodes
  - Output: Individual link utilization
- Models contention and backpressure:



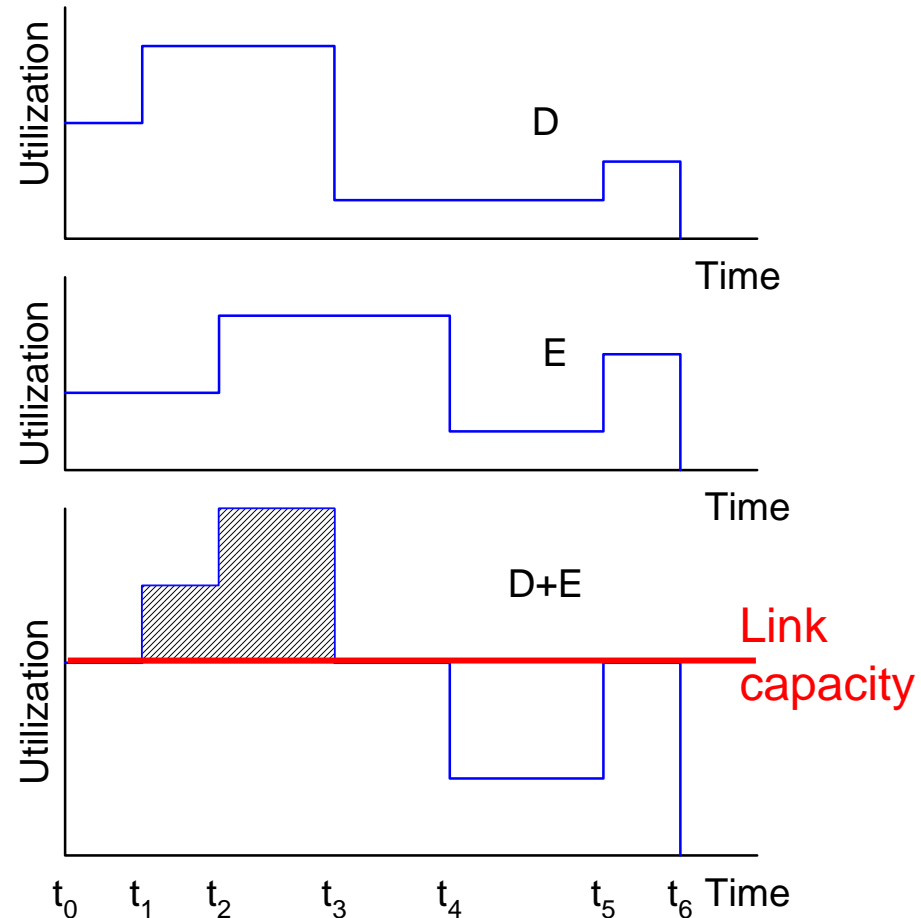
# LUNA: High-level network power analysis

- Link utilization as proxy for node power
  - Input: Message flows between nodes
  - Output: Individual link utilization
- Models contention and backpressure



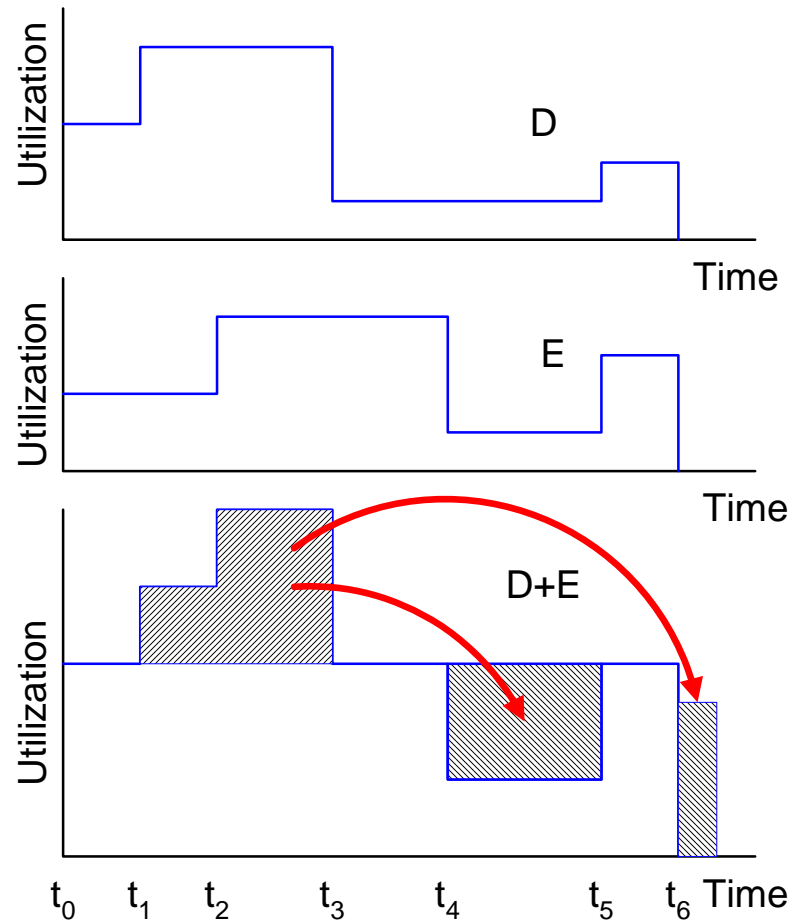
# LUNA: High-level network power analysis

- Link utilization as proxy for node power
  - Input: Message flows between nodes
  - Output: Individual link utilization
- Models contention and backpressure



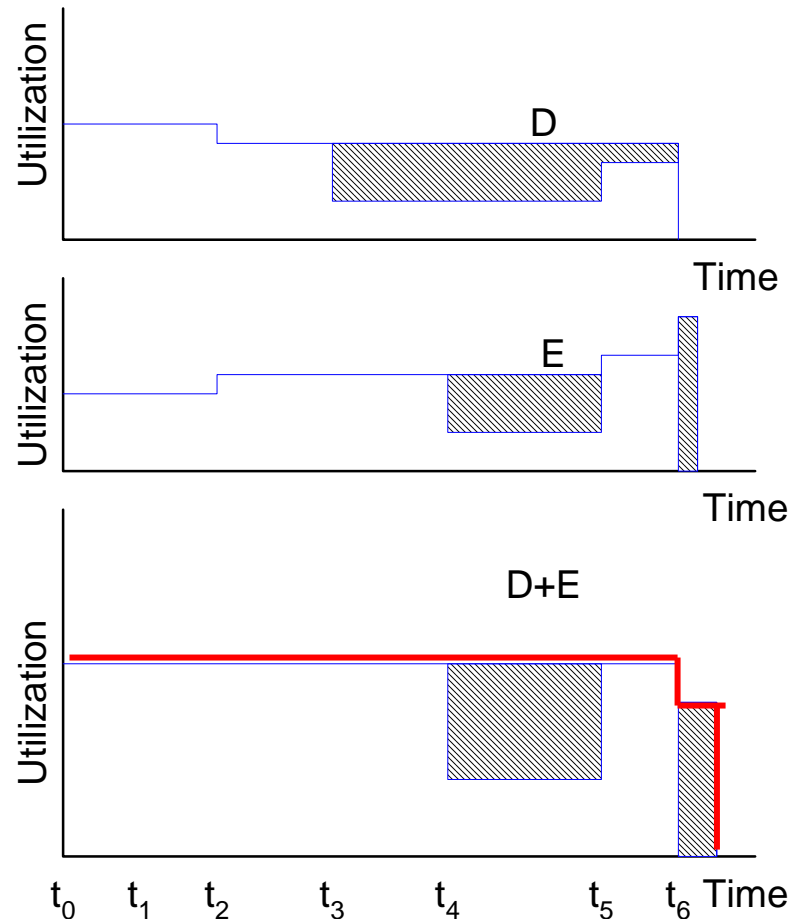
# LUNA: High-level network power analysis

- Link utilization as proxy for node power
  - Input: Message flows between nodes
  - Output: Individual link utilization
- Models contention and backpressure



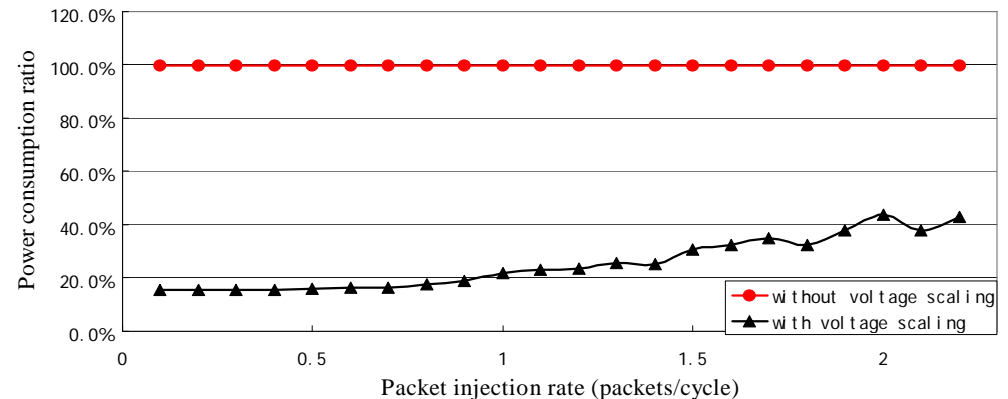
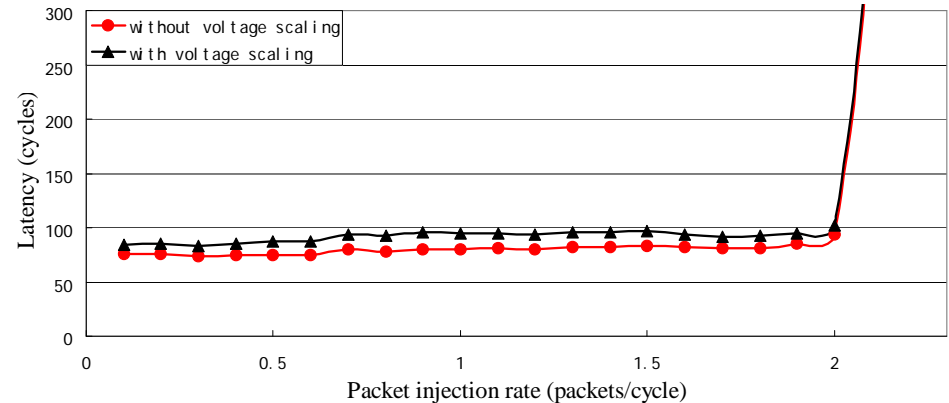
# LUNA: High-level network power analysis

- Link utilization as proxy for node power
  - Input: Message flows between nodes
  - Output: Individual link utilization
- Models contention and backpressure
- Validation: <9% relative error against ORION with up to 2 orders of magnitude speedup
- Status:
  - Used for rapid design space exploration of networks (Intel), compiler power management (Kandemir&Irwin, PLDI'06)
  - Ongoing refinement and validation with Intel
  - 50+ downloads



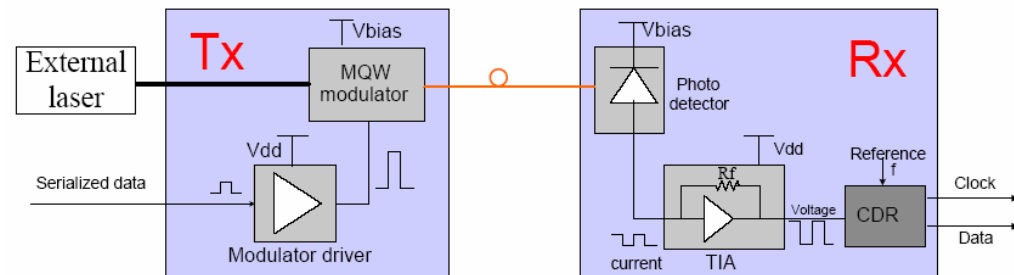
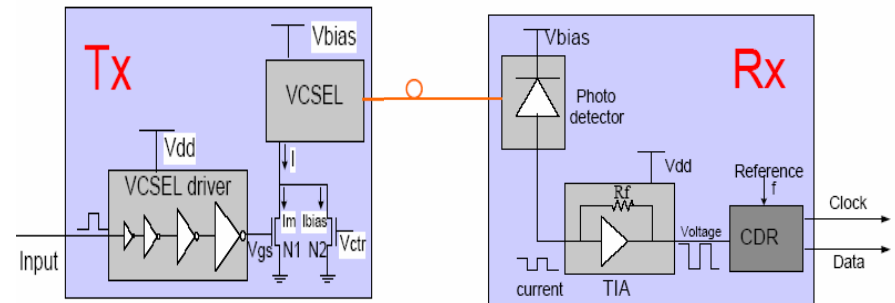
# Dynamic voltage scalable networks

1. Explore potential power savings and latency impact



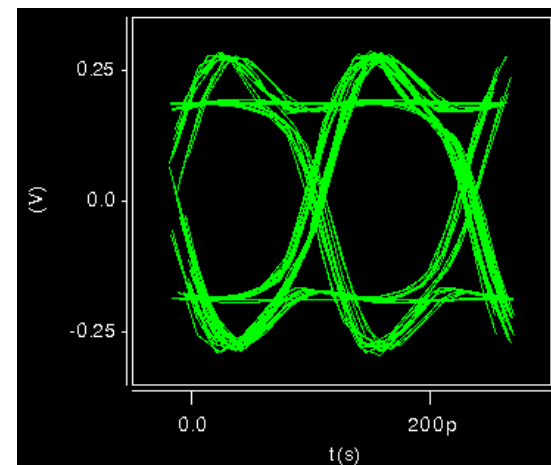
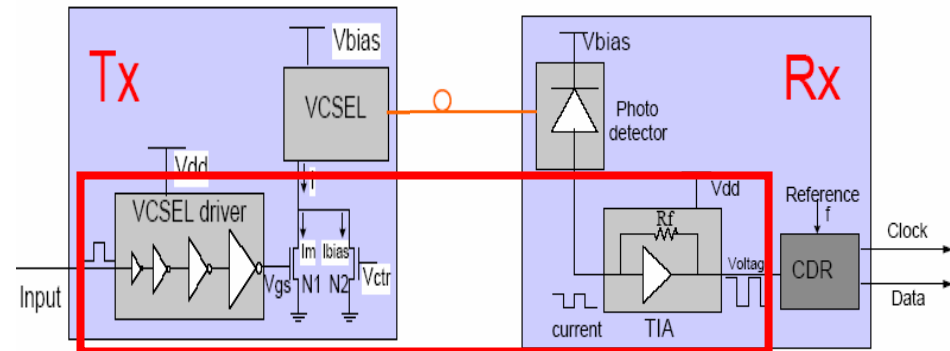
# Dynamic voltage scalable networks

1. Explore potential power savings and latency impact
2. Design DVS opto-electronic networked system



# Dynamic voltage scalable networks

1. Explore potential power savings and latency impact
2. Design DVS opto-electronic networked system
3. DVS link design
  - Variable-voltage front-end
  - UMC 130nm: 0.7-1.2V, 4-8Gb/s, 6.7-40.8 mW



---

# Outline

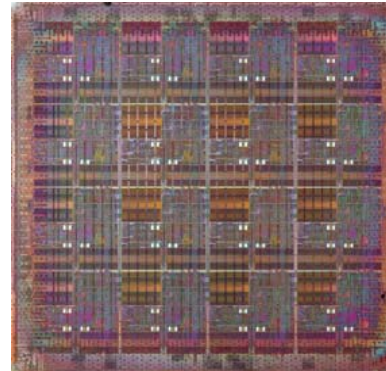
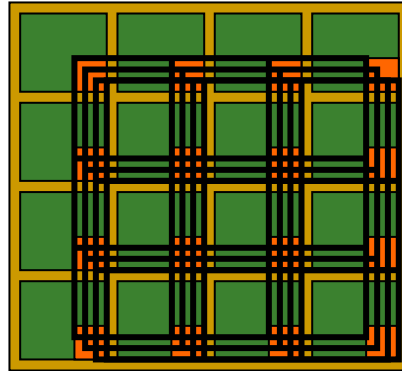
- Brief survey of group's research
    - Power-driven network design tools
      - ORION: Architectural network power models
      - LUNA: High-level network power analysis
    - Power-aware networks
      - Dynamic voltage scalable networks
  - Thermal modeling and management of on-chip networks
  - Next: Network-driven computing
-

---

# Thermal modeling and management of on-chip networks

---

# Power and thermal-efficient on-chip networks: A motivating example



## **MIT Raw (0.18um, 300MHz)**

First fabricated networked CMP

### Core:

8-stage in-order single-issue pipeline

4-stage single-precision FPU

32KB cache

32KB software-managed cache

### On-chip interconnection network:

Four 32-bit 4x4 networks

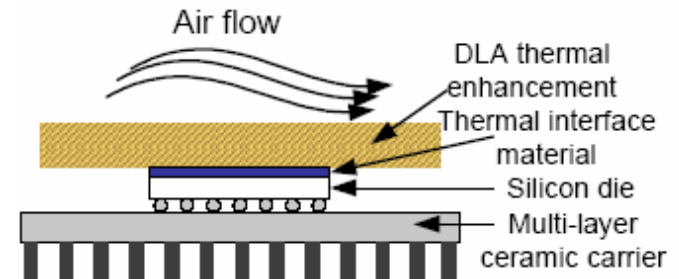
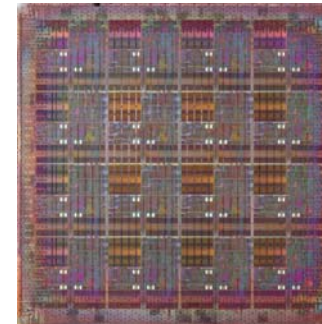
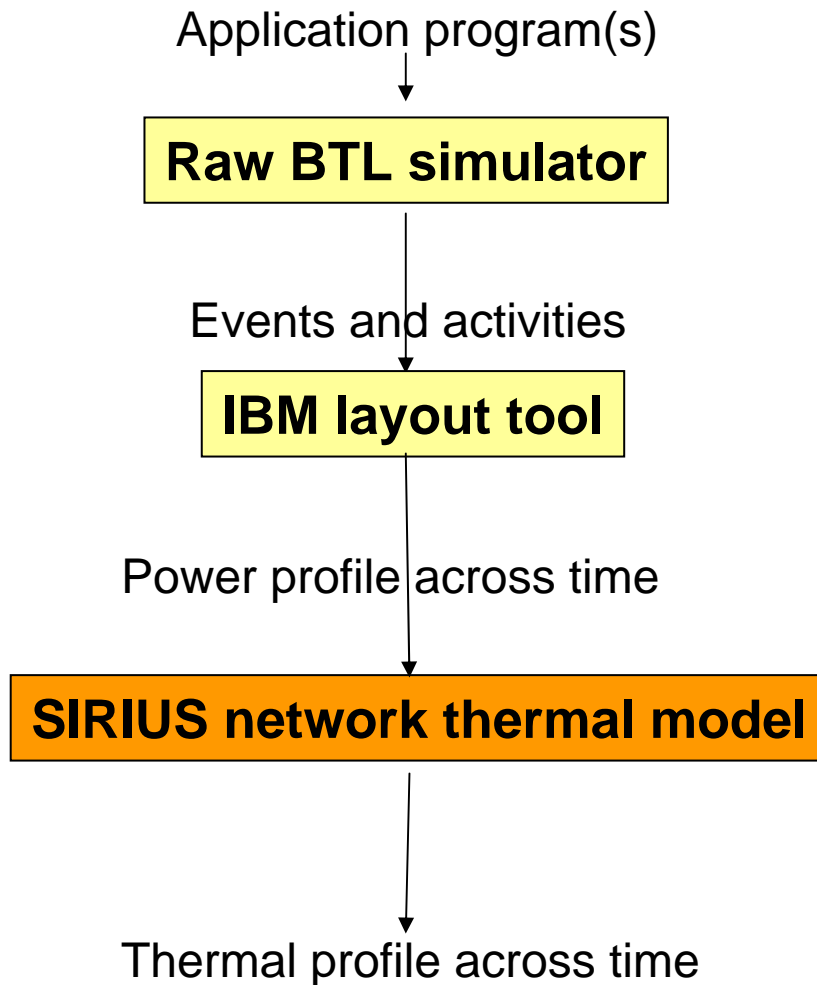
2 static (8KB software-managed I\$)

2 dynamic

Average: 7.2W, Peak: 14.8W

36% of avg chip power

# Methodology



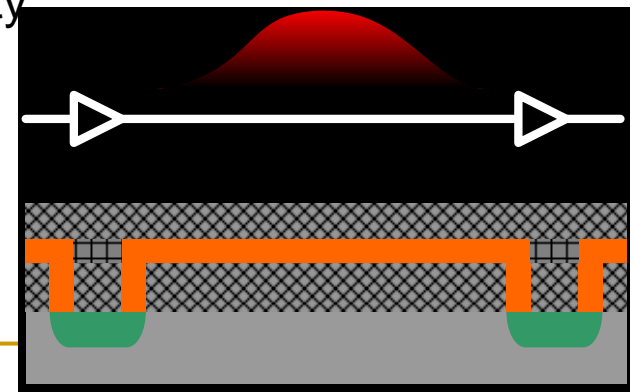
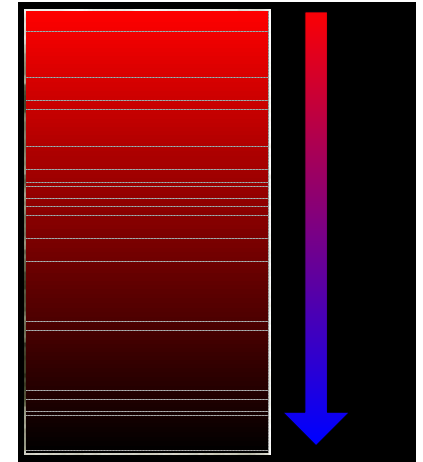
Why a network thermal model?

- Lateral heat spreading is more critical in an on-chip network
- Need to model on-chip interconnects/links



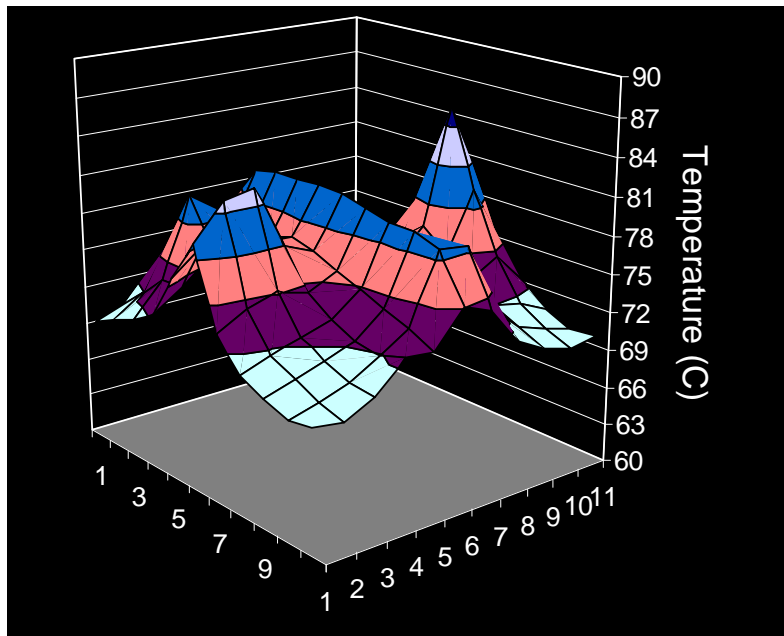
# SIRIUS: Modeling of On-Chip Links

- Thermal impact of on-chip link circuitry
  - Reduced metal pitch and increased metal layers
  - High thermal resistance of silicon dioxide layers
  - High thermal resistance of low-k insulator materials
- Thermal modeling of on-chip link circuitry
  - Thermal impact of buffers
    - Increased power consumption hence silicon temperature
    - Copper vias have high thermal conductivity
  - Buffer insertion estimation
  - Temperature profile estimation

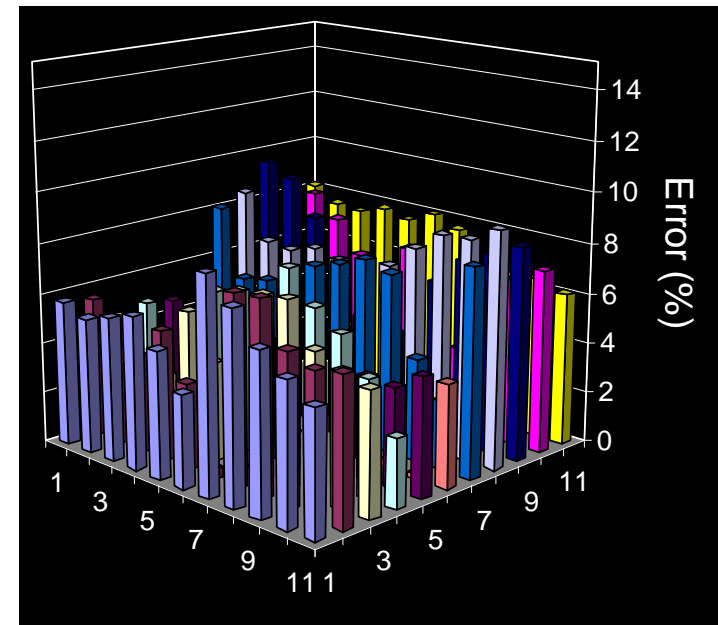


# Thermal Model Validation

- Modeled actual chip design from IBM
  - In-house finite-element based thermal simulator
  - Our model: [70.2, 85.4] °C; IBM: [73.2, 87.8]°C

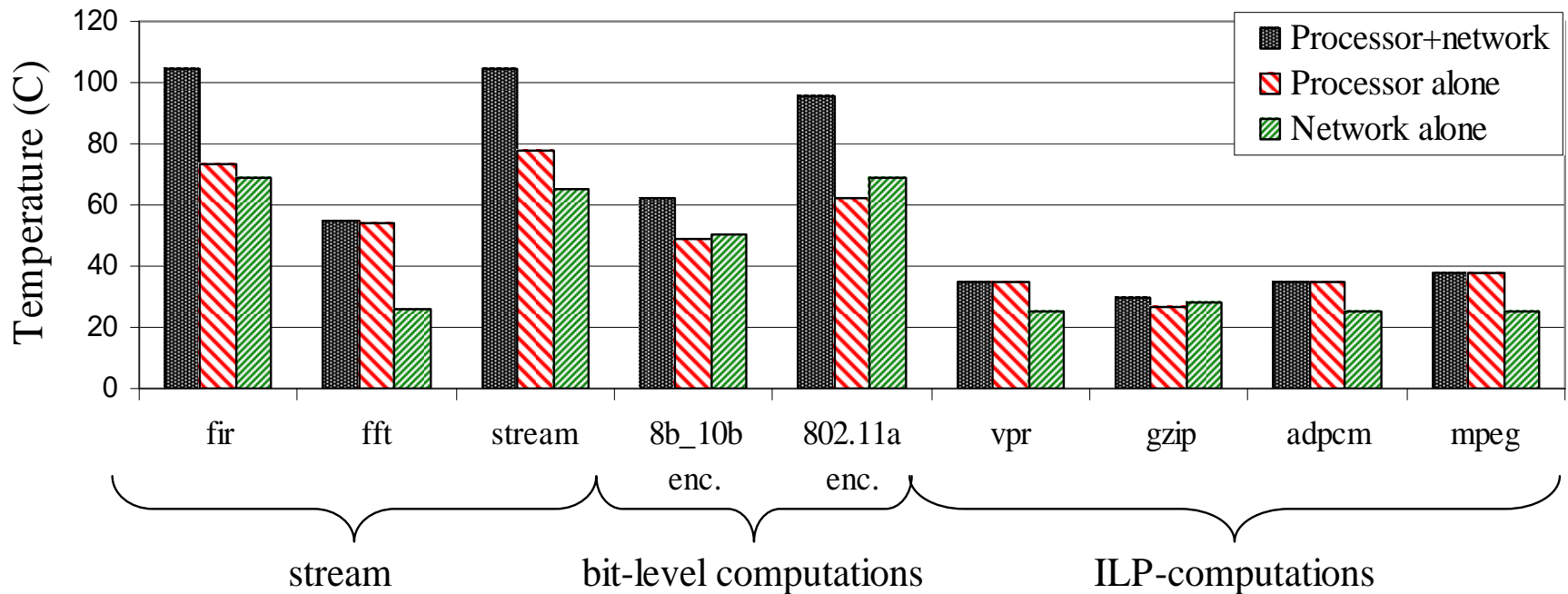


Chip temperature profile using our thermal model



Estimation error

# Thermal characterization of RAW chip

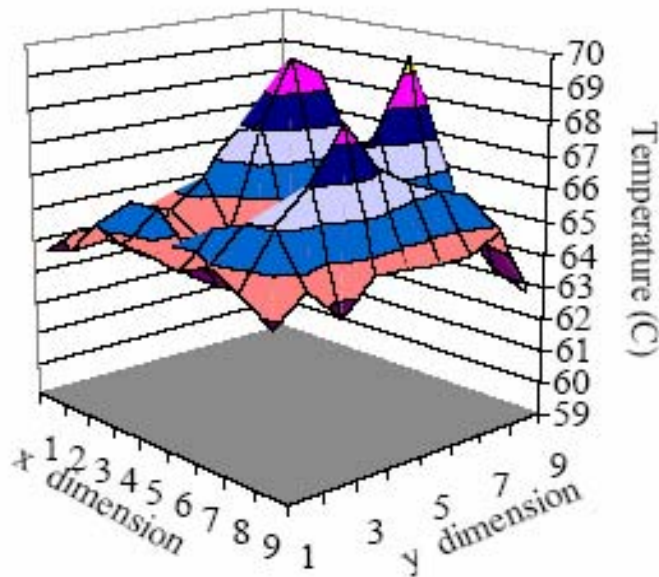


**Networks are significant power/thermal contributors**  
**So, how can we target this?**

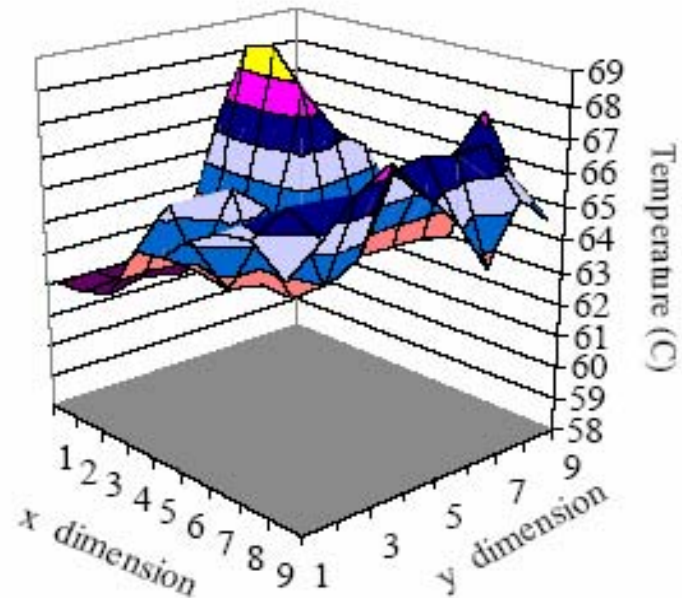
# Run-time network thermal management to guarantee safe on-line operation

High spatial and temporal variance in network temperature

Temperature at time t1



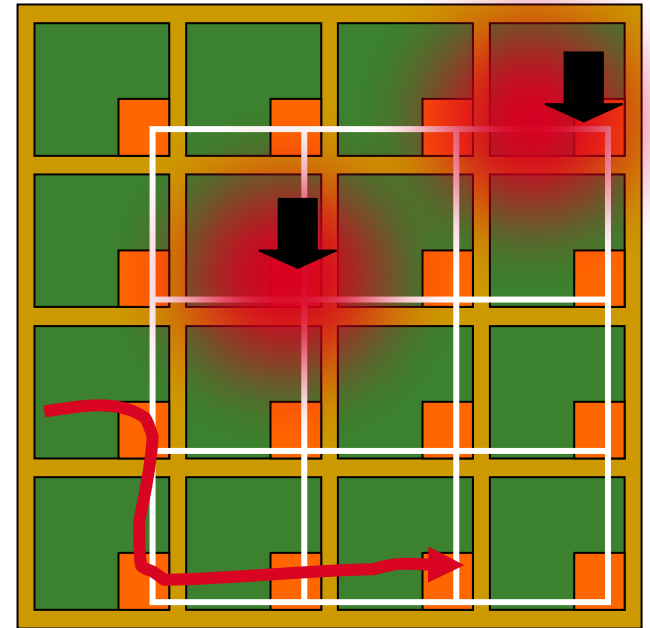
Temperature at time t2



***Thermal design for worst-case no longer cost-effective***

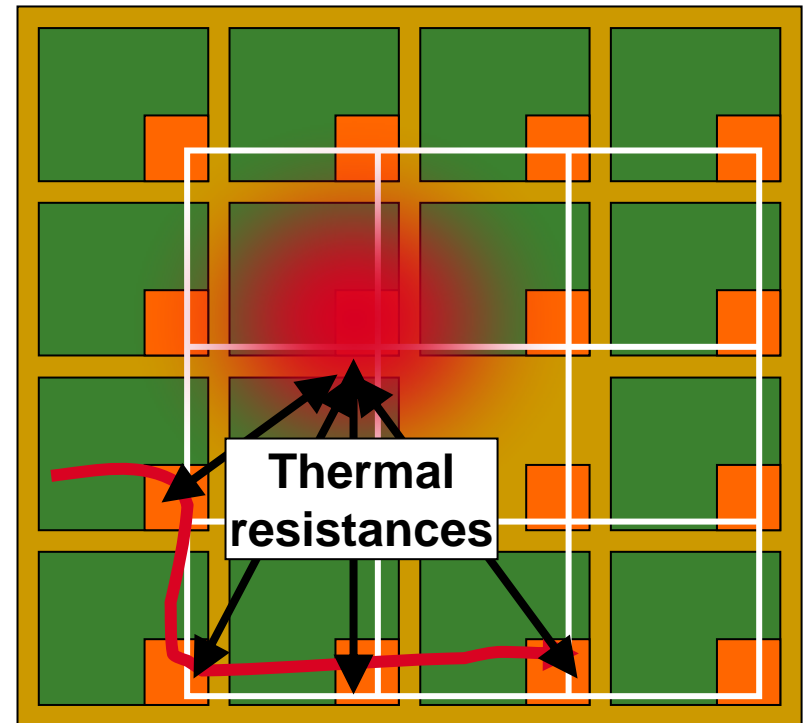
# ThermalHerd: Let's dynamically steer network traffic to avoid thermal hotspots

1. Temperature monitoring
2. Traffic estimation & prediction
3. Before emergencies
  - Proactive thermal-aware routing
4. Upon emergencies
  - Distributed traffic throttling
  - Reactive thermal-aware routing



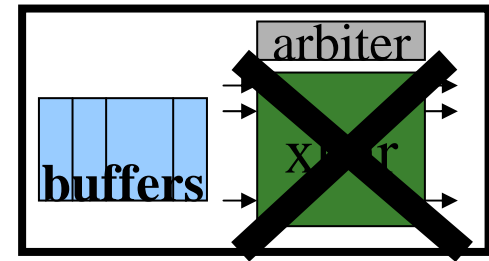
# ThermalHerd: Thermal Correlation-Based Routing

- Chooses a route based on how thermally correlated it is with the hotspot
  - Hotspot notification messages
  - Pre-computed thermal resistance matrix (based on *Sirius* thermal model) stored at each router
  - Choose minimal path where thermal correlation between source and every hop along the path is  $<$  threshold
- Threshold is less aggressive prior to thermal emergencies



# ThermalHerd: Temperature-Aware Traffic Throttling

- How much to throttle?
  - Each router is assigned a quota: number of packets it is allowed to process within a time window
  - Quota reduced exponentially as temperature rises
  - Quota prioritizes local traffic over passing-by traffic
- How to throttle?
  - Disable switch allocation



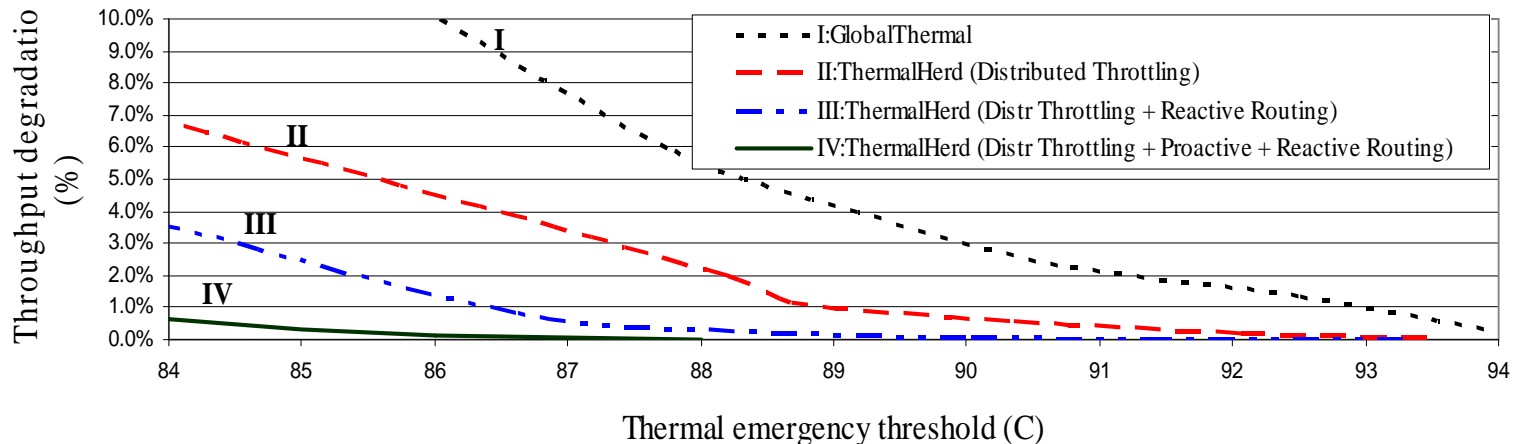
# ThermalHerd evaluation (Austin TRIPS traces for 16 benchmarks)

Peak temp. without ThermalHerd (94 degrees)

Peak temp. constraint  
Peak temp. as managed  
by ThermalHerd

$T_T$ ( $^{\circ}\text{C}$ )	89	87	85	83	82	79	77	75
$T_A$ ( $^{\circ}\text{C}$ )	86.2	86.1	84.1	82.5	81.3	78.8	76.5	74.7

**ThermalHerd effectively regulates peak temperature...**



**....with little performance degradation**

---

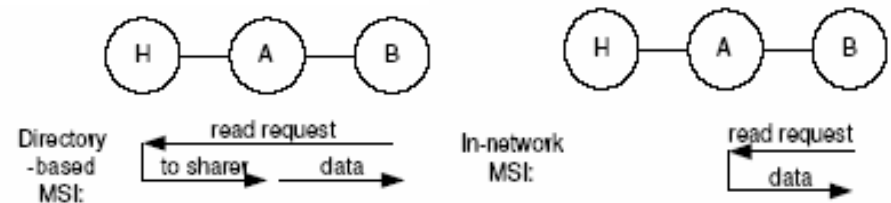
# Outline

- Brief survey of group's research
  - Power-driven network design tools
    - ORION: Architectural network power models
    - LUNA: High-level network power analysis
  - Power-aware networks
    - Dynamic voltage scalable networks
- Thermal modeling and management of on-chip networks
- **Next: Network-driven computing**

# Network-driven Computing

- Network-Driven Computing
  - Where the on-chip network no longer handles just communication, but also drives all coordination between cores
- Why?
  - *Fast access*
  - *In-transit adaptation*
  - *Scalable complexity*

- E.g. in-network cache coherence (poster)



- Sharers kept track of using trees within network routers: Network routes requests to nearest copy.
- Original directory protocol:
  - Requestor (B) to Home Directory (H) to Sharer (A)
- In-network coherence:
  - Enroute from Requestor (B) to Directory (H), network routes request to nearest Sharer (A) and back.
- Scalable:
  - Lessen global traffic
  - Storage overhead scales with number of ports instead of number of cores