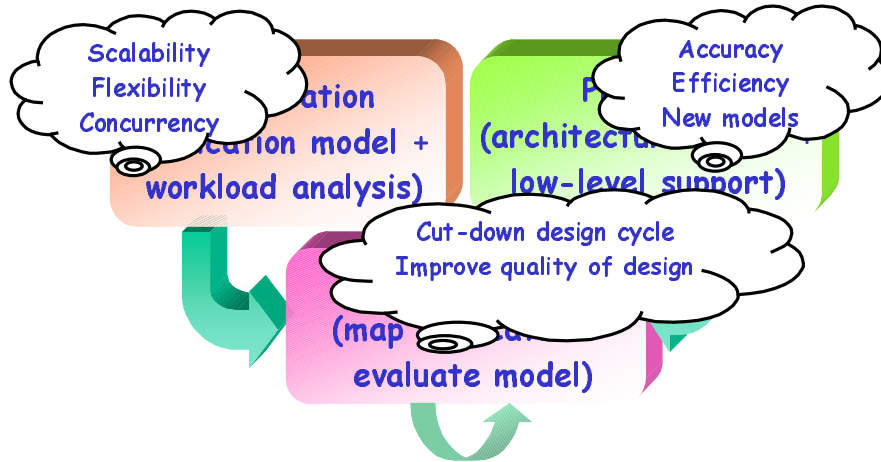# Formal Models for Power/Performance Analysis in System-Level Design

Radu Marculescu

Department of Electrical & Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

RM   1

Outline

- Motivation

- Stochastic Automata Networks (SANs)
  - Application Models
  - Architecture Models
  - Mapping

- Experimental results
  - Implications in the design process

RM   2

The Big Picture

Scalability
Flexibility
Concurrency

ation
cation model +
workload analysis)

Accuracy
Efficiency
New models

(architectu
low-level support)

Cut-down design cycle
Improve quality of design

(map
evaluate model)

The SAN Modeling Paradigm

- Idea
  - Model construction {PG, rates, communication}
    - Application is a process graph w/ processes active concurrently
    - Process graph translates into a network of automata
  - Model evaluation
    - Use tensor products (representations remain compact!)
    - Used to compute latency, utilization, response time, ...
      - *True* rates vs. specified rates

- Interactions among automata
  - Synchronizing transitions
    - May alter the state of possible many automata
      - Transitions that are not synchronized are local
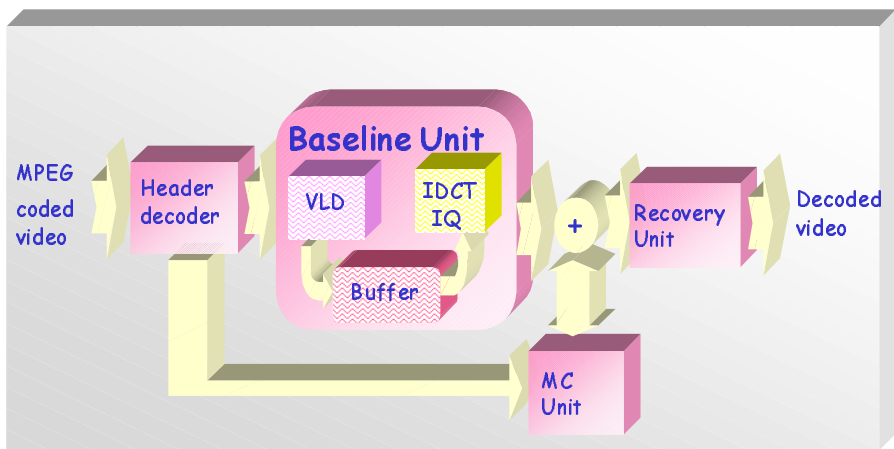 - Functional transitions
    - Affect the state of a single automaton

Steady-State Regime

- Global descriptor: $Q = \displaystyle\sum_{j=1}^{2E+N} \bigotimes_{i=1}^{N} Q_j^{(i)}$

  - $\pi \cdot Q = 0$

    $\sum_i \pi_i = 1$

  - Note: we want to avoid the explicit construction of Q!

- Use iterative methods

  - Complexity reduces to

    $$\prod_{i=1}^{N} n_i \times \sum_{i=1}^{N} n_i$$
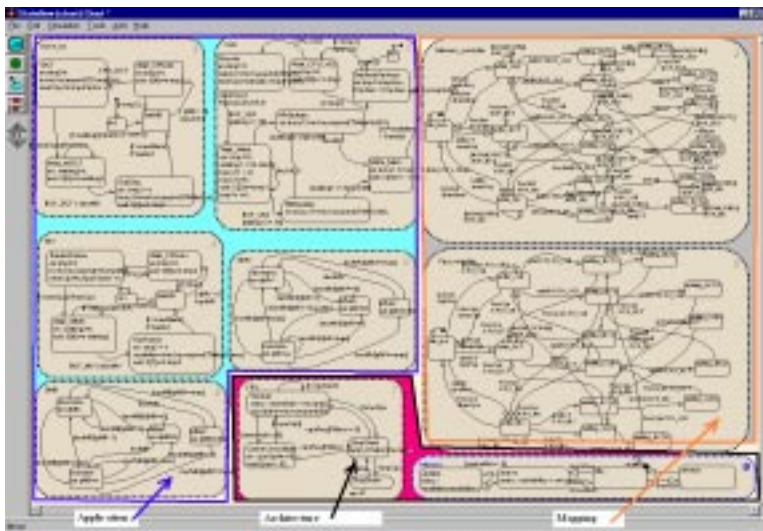
RM
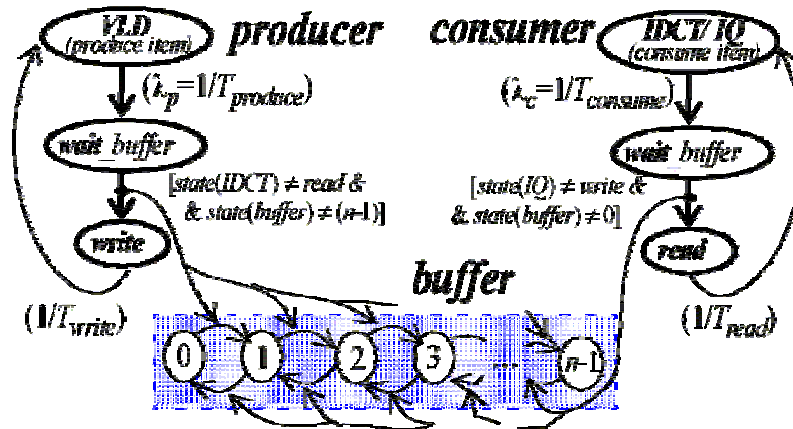
5

The Ping-Pong Protocol
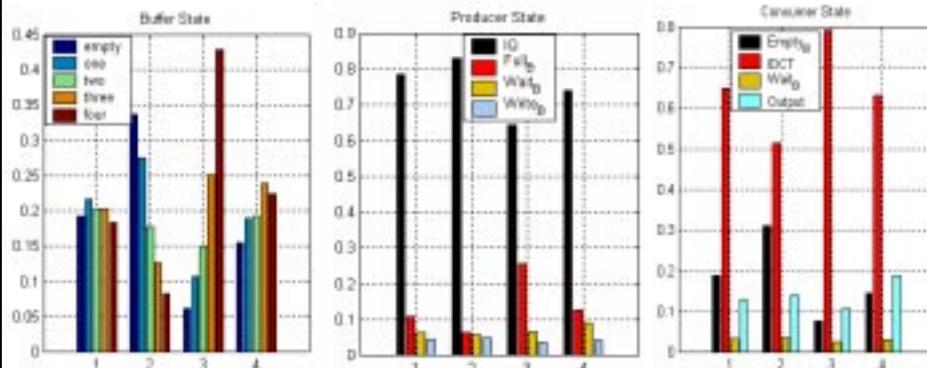

RM

6
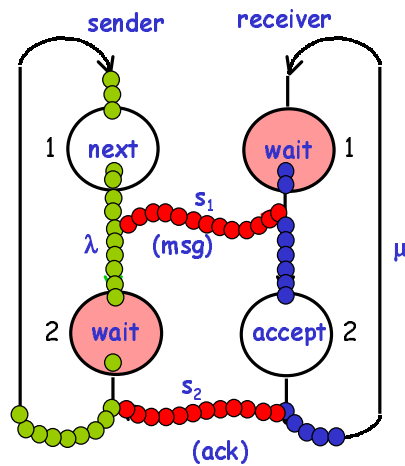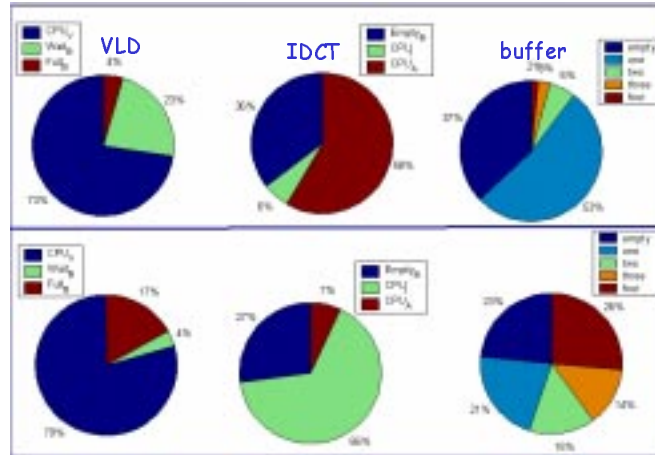
MPEG-2 Decoder

Stateflow (screen shot)

Modeling the Application

Dual CPU Case

MPEG-2 Application Analysis

Mapping 1

Mapping Processes to Processors
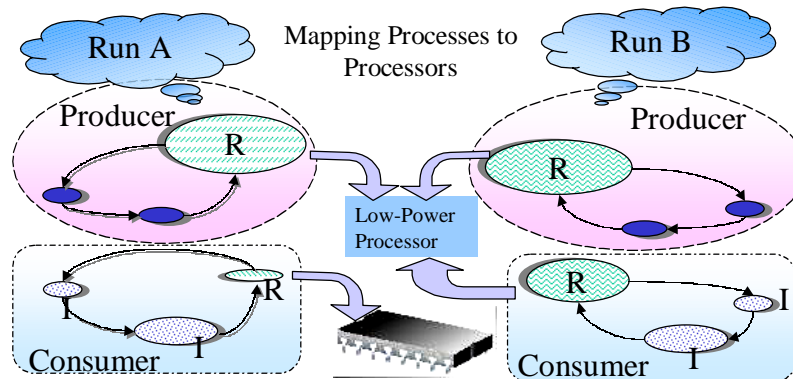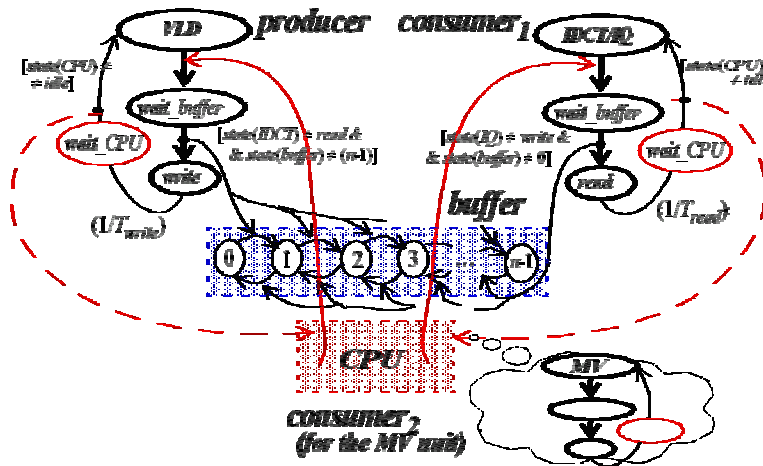
Modeling the Architecture

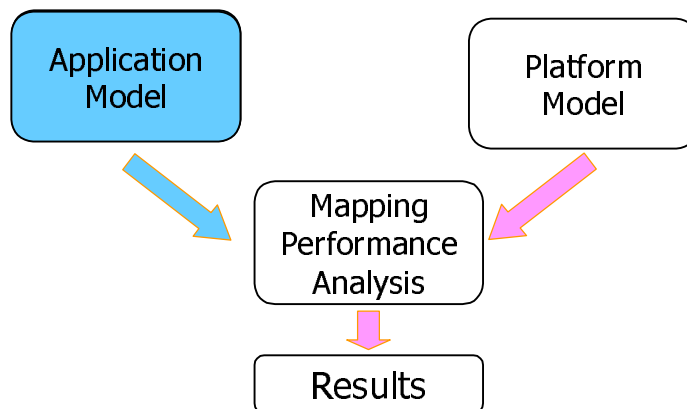Model of the memory

Model of the CPU
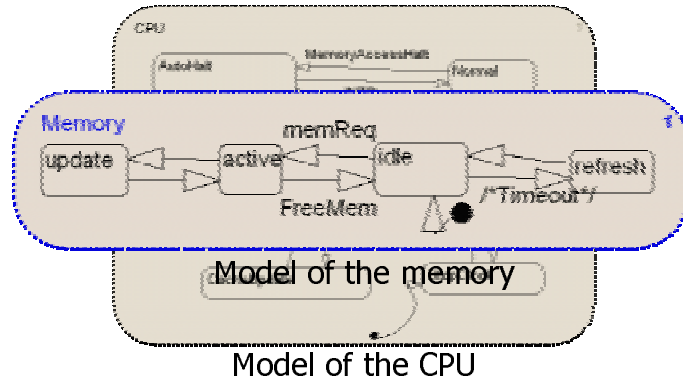
Putting Everything Together

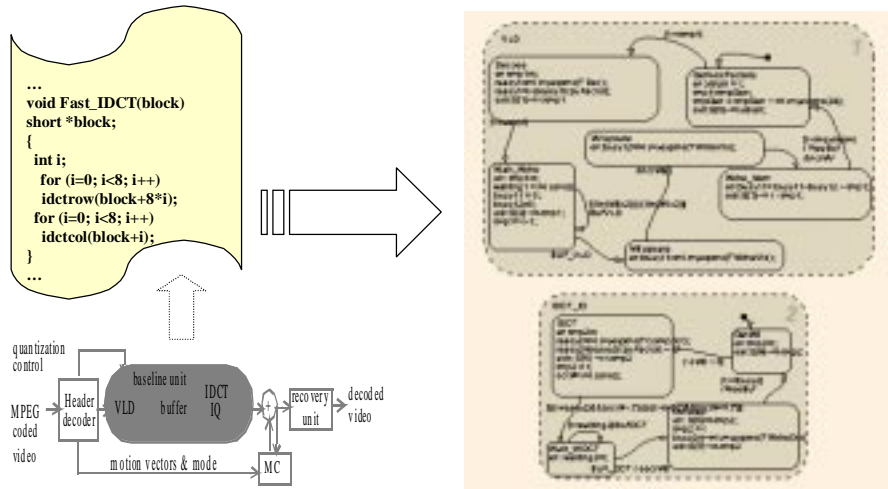# Outline

✓ Motivation

✓ Stochastic Automata Networks (SANs)
- Application Models
- Architecture Models
- Mapping

- Experimental results
  - Implications in the design process

Experimental Setup

Application Modeling
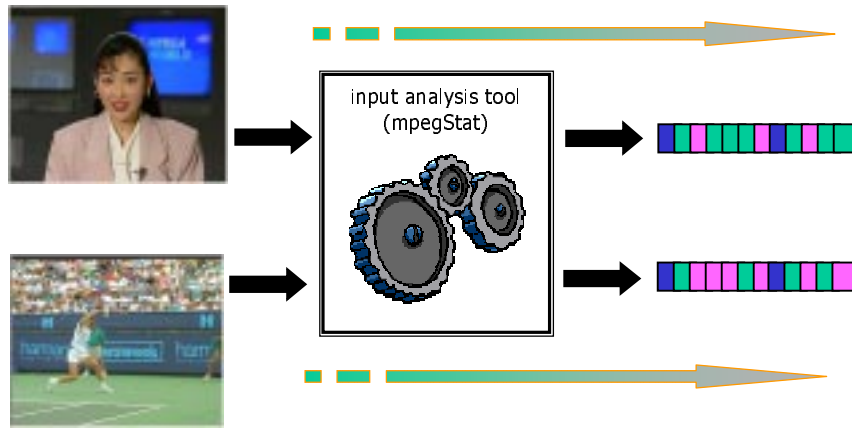


```
…
void Fast_IDCT(block)
short *block;
{
  int i;
   for (i=0; i<8; i++)
   idctrow(block+8*i);
  for (i=0; i<8; i++)
   idctcol(block+i);
}
…
```

RM

17

Environment Modeling

input analysis tool
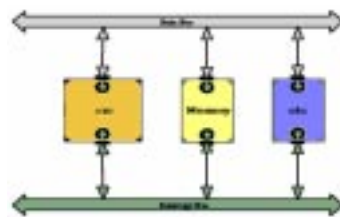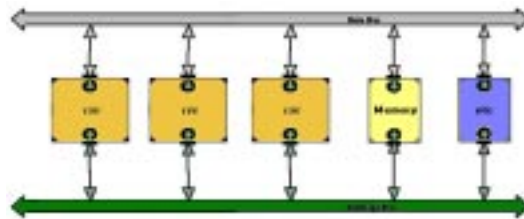(mpegStat)

RM

18

9

Platform Models

infinite resources

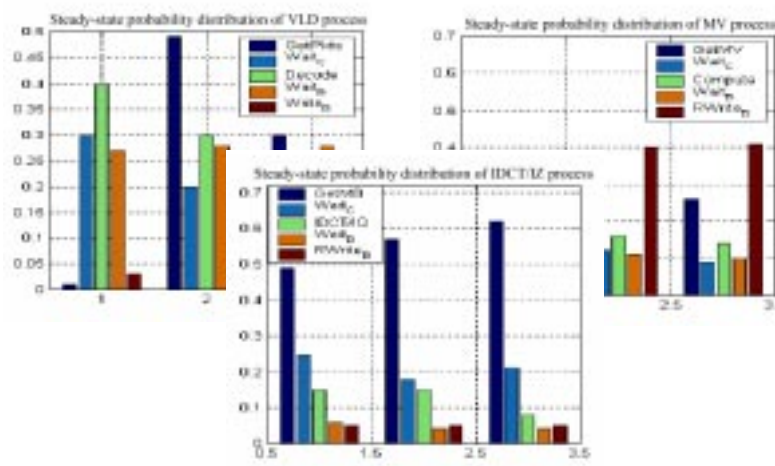**finite resources**
CPU-Speed:
$\{ f_0 , 2f_0 , 3f_0 \}$

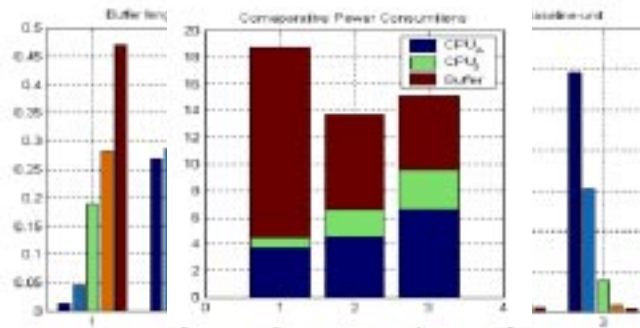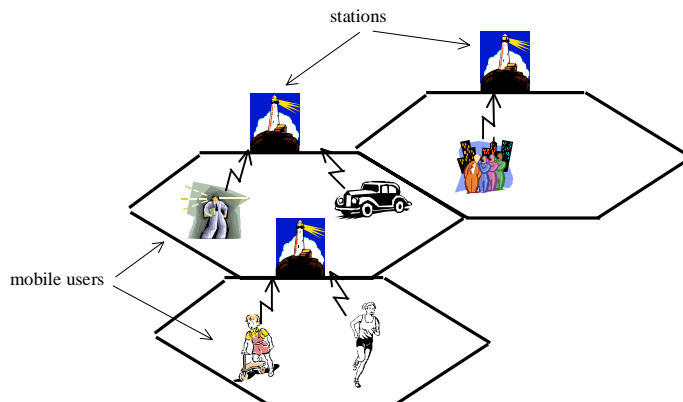# Results

10

Results

$$P^{(k)} = \sum_{all\ i} \pi_i \cdot P_i + \sum_{all\ i,j} \lambda_{ij} \cdot P_{ij}$$

Pico-cell Environment

stations

mobile users

The Encoder – Decoder Pair

ENCODER

P   C/P

queue

wireless (lossy) channel

DECODER

P/C   C   display
(VLD)   (IDCT)

queue

Channel Modeling

∞ Buffer

Encoder   Decoder   Application level

Finite Buffer

Encoder [Buffer-Tx] Tx   Rx [Buffer-Rx] Decoder [HW] [scheduler]   Application mapped to Hardware

Ideal Channel

Communication Error

Encoder [Buffer-Tx] Tx   Error Model   Rx [Buffer-Rx] Decoder [HW] [scheduler]

Real Channel
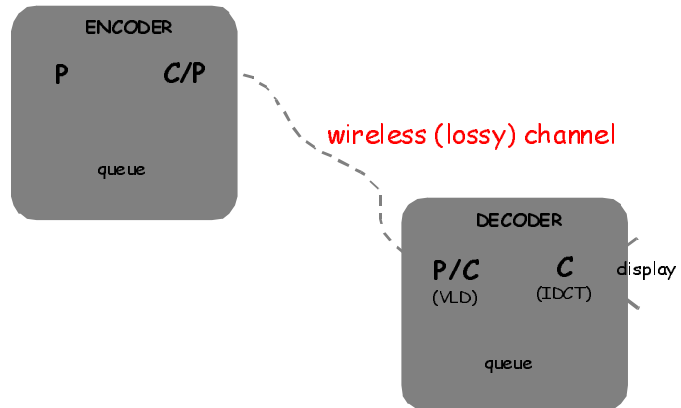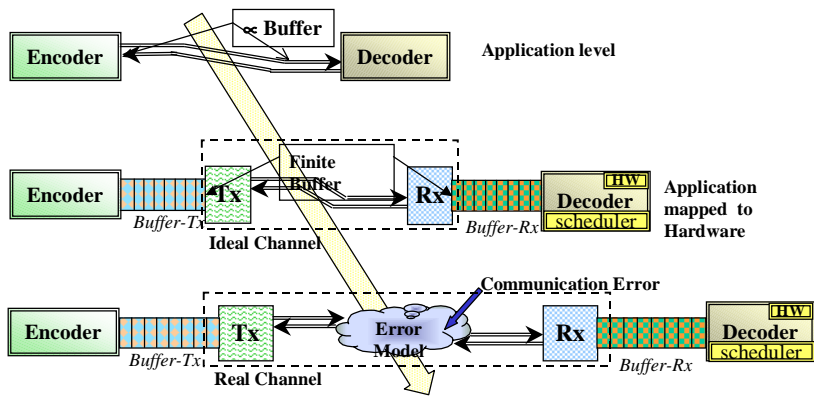
Source-Channel-Receiver
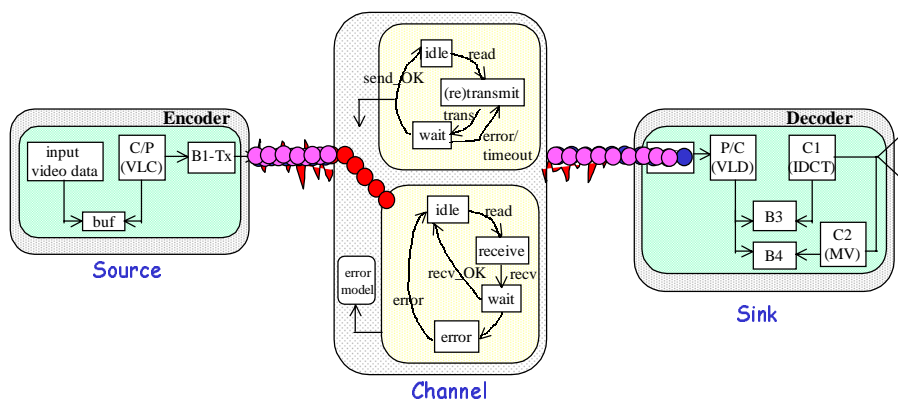
Buffer Length Variation

Power and Latency

Power Calculation in Watts — PowerE, PowerL

Latency — channel, source
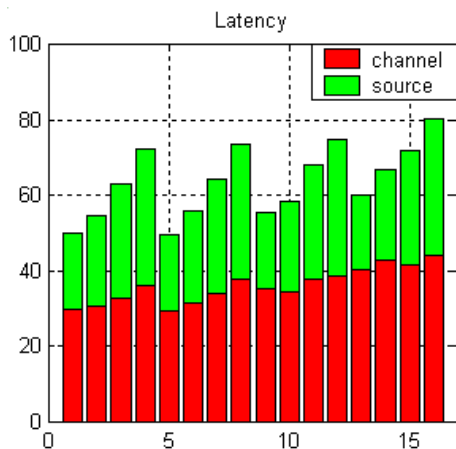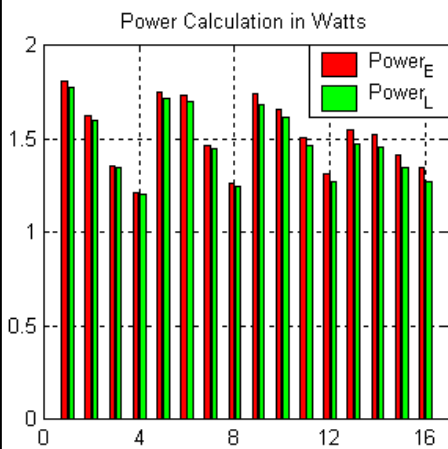
MPEG Coder with Prioritization

DCT — Q — PC — VLC → Bit stream (low priority)

Q^{-1}

DCT^{-1}

VLC → Bit stream (high priority)

MV — FM

MPEG coder — LP / HP — Packetizer — Multiplexer

HP cells

LP cells
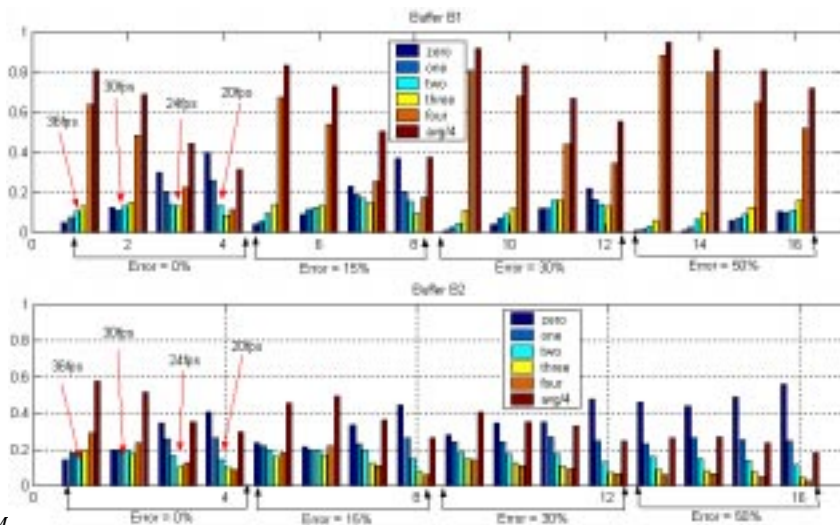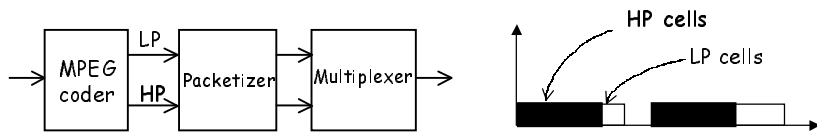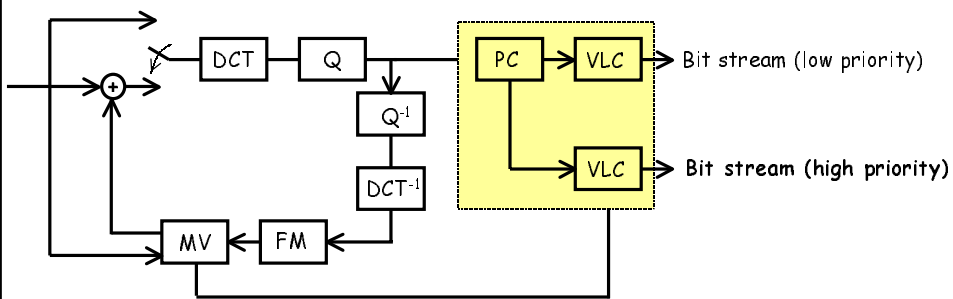
Complete Application Modeling

ENCODER

P   C/P

queue

wireless (lossy) channel

DECODER

P/C   C   display
(VLD)   (IDCT)

queue

(on-chip)
scheduler
(centralized/distributed)
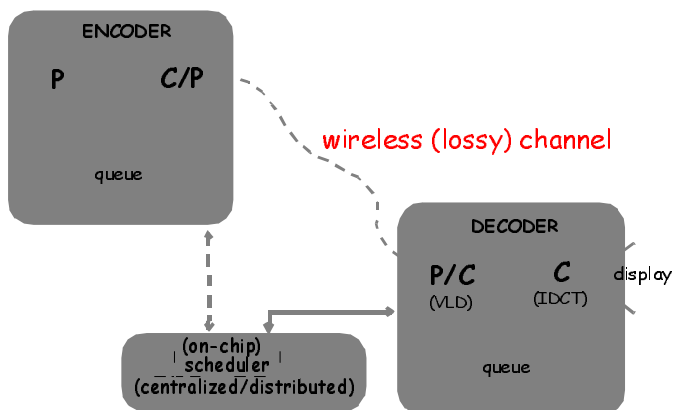
Applications that Communicate over Wired and Wireless Channels

Application 1

P   C

queue

etc...

channel_1
(lossless)

channel_2 (lossy)

Application 2

C   P

queue

Application 3

C   P

queue

channel_3 (lossy)

Long(er) Term Vision

- Concurrency
- Granularity/complexity
- Analysis

Application

- Scheme
- Topology
- Mapping
- Scheduling

- PE Selection
- Mapping
- Scheduling
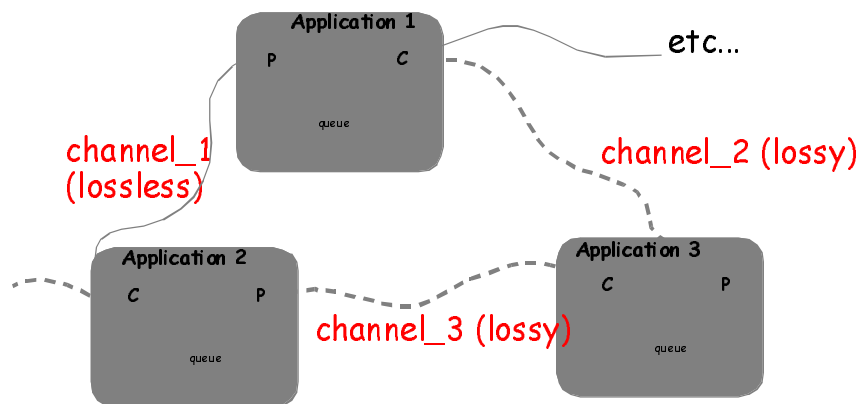
Architecture   **Communication**
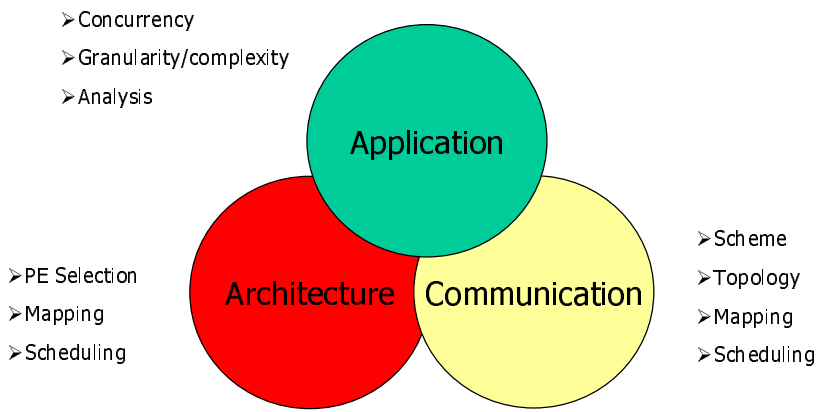
Integration of the Analysis Module within Metropolis Framework

Design of Function Processes   **Design of Communication Media**   **Design of Architecture Components**

Metropolis Infrastructure

Metropolis Point Tools: Synthesis/Refinement   **Metropolis Point Tools: Analysis/Verification**

Analysis

abstract mapping + performance numbers

scheduling policy + adaptation mechanism
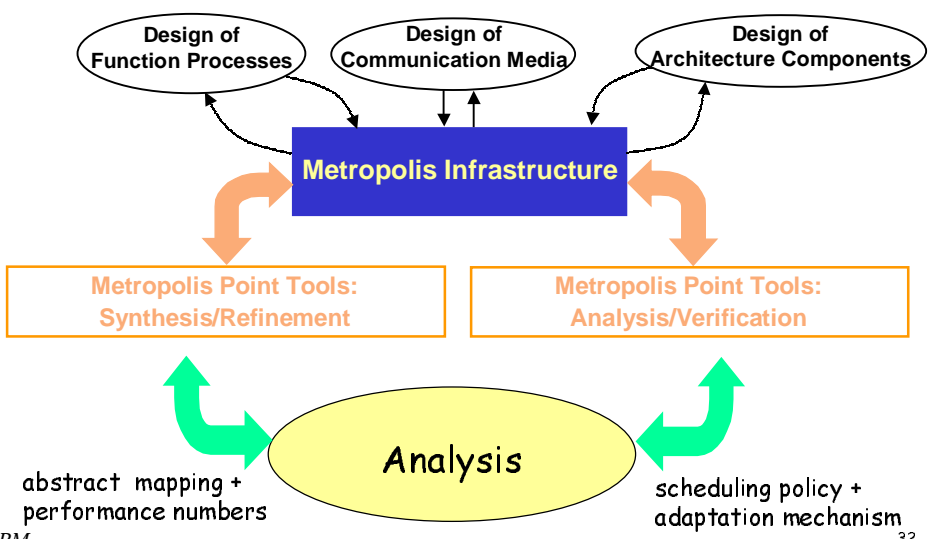
Summary

- ■ Ideas
  - ◆ Probabilistic framework
    - ➢ formalization is possible!
  - ◆ Analysis driven by environment
  - ◆ Application/architecture modeling
    - ➢ Scalability
    - ➢ Flexibility
    - ➢ Concurrency

- ■ Ongoing work
  - ◆ Optimization of the analytical model
  - ◆ Scheduling effects
  - ◆ On-chip communication analysis
  - ◆ Speeding-up simulation techniques

RM   33

17