Design Space Exploration of Network Processor Architectures

ECE 697J

December 3rd, 2002

Introduction

- Network processor architectures have many choices
  - Number of processors
  - Size of memory
  - Type of interconnect
  - Co-processors
- How can the optimal configuration be found?
  - Need to explore design space
- Simulations are too system-specific and limited
- Analytic performance modeling
  - Performance expressed as function of NP configuration and workload

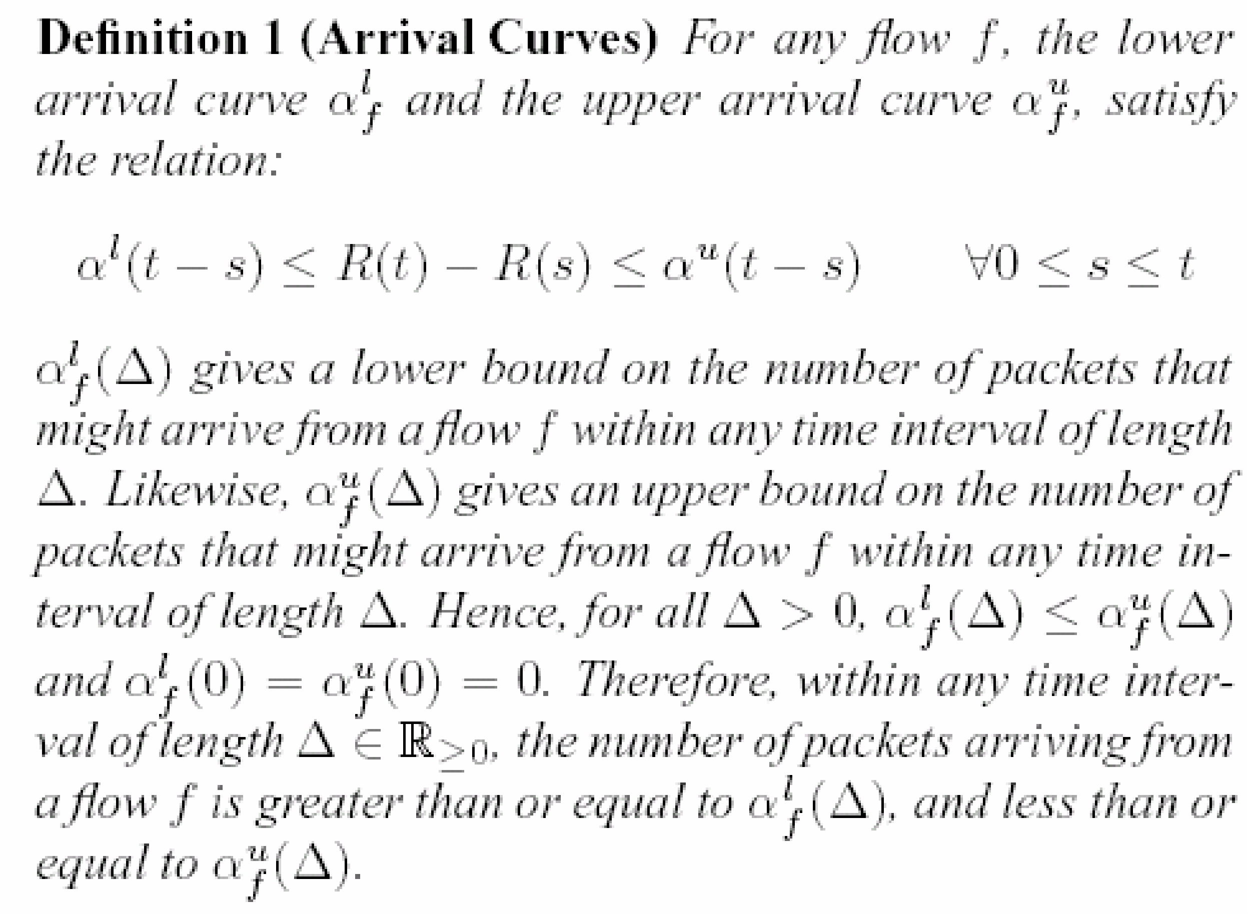
System Components

- Model assumes SOC with:
  - Multiple processor cores
  - Micro-engines
  - Dedicated hardware (co-processors)
  - Memory units, caches
  - Interconnects
  - I/O interfaces
- Very general, matches most NPs

Analytic Model

- Performance model
  - Task (= processing requirements)
  - Resources (= NP resources)
  - Mapping of tasks to resources
  - Traffic (= arrival curves)
- Optimization (= cost function)
  - Chip area
  - On-chip memory
  - Performance
- Other properties
  - Delay
  - Throughput

Arrival Curves



Definition 1 (Arrival Curves) *For any flow f, the lower arrival curve α^l_f and the upper arrival curve α^u_f, satisfy the relation:*

$$\alpha^l(t-s) \leq R(t) - R(s) \leq \alpha^u(t-s) \qquad \forall 0 \leq s \leq t$$

$\alpha^l_f(\Delta)$ gives a lower bound on the number of packets that might arrive from a flow f within any time interval of length Δ. Likewise, $\alpha^u_f(\Delta)$ gives an upper bound on the number of packets that might arrive from a flow f within any time interval of length Δ. Hence, for all $\Delta > 0$, $\alpha^l_f(\Delta) \leq \alpha^u_f(\Delta)$ and $\alpha^l_f(0) = \alpha^u_f(0) = 0$. Therefore, within any time interval of length $\Delta \in \mathbb{R}_{\geq 0}$, the number of packets arriving from a flow f is greater than or equal to $\alpha^l_f(\Delta)$, and less than or equal to $\alpha^u_f(\Delta)$.

Arrival Curves

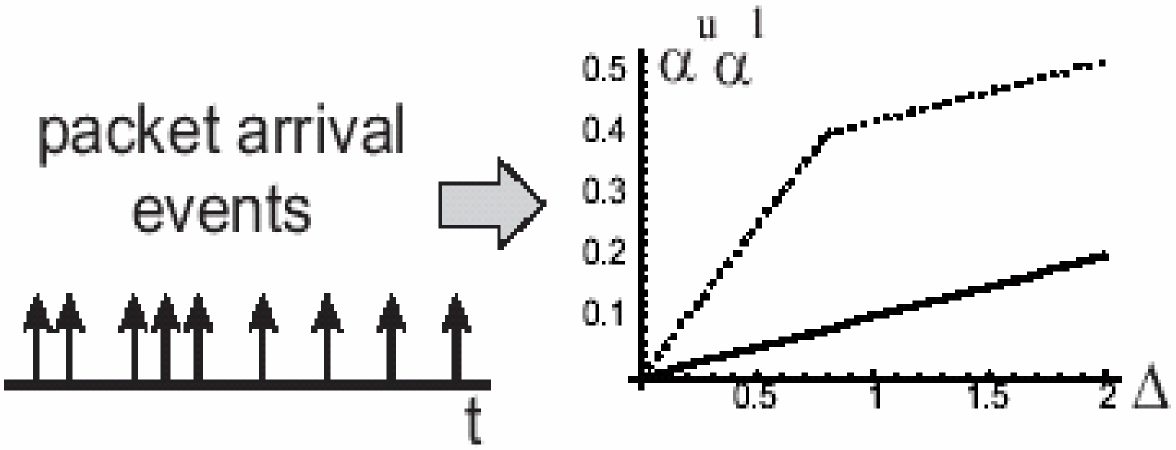
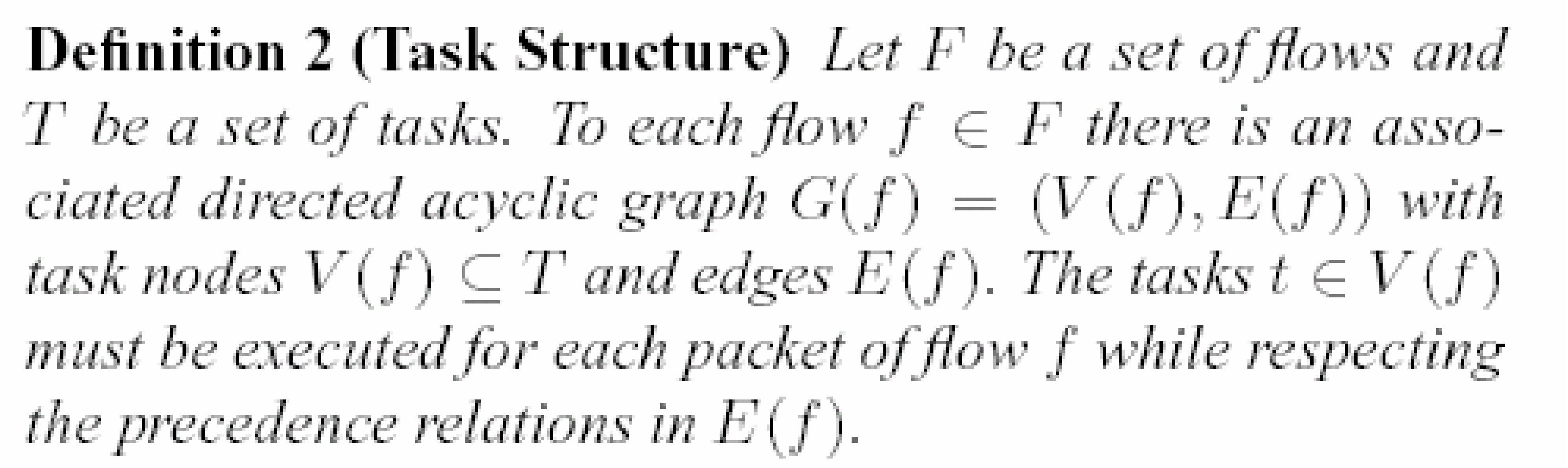


Figure 1. Representation of arrival curves.

Task Structure



Definition 2 (Task Structure) *Let F be a set of flows and T be a set of tasks. To each flow $f \in F$ there is an associated directed acyclic graph $G(f) = (V(f), E(f))$ with task nodes $V(f) \subseteq T$ and edges $E(f)$. The tasks $t \in V(f)$ must be executed for each packet of flow f while respecting the precedence relations in $E(f)$.*

Task Structure

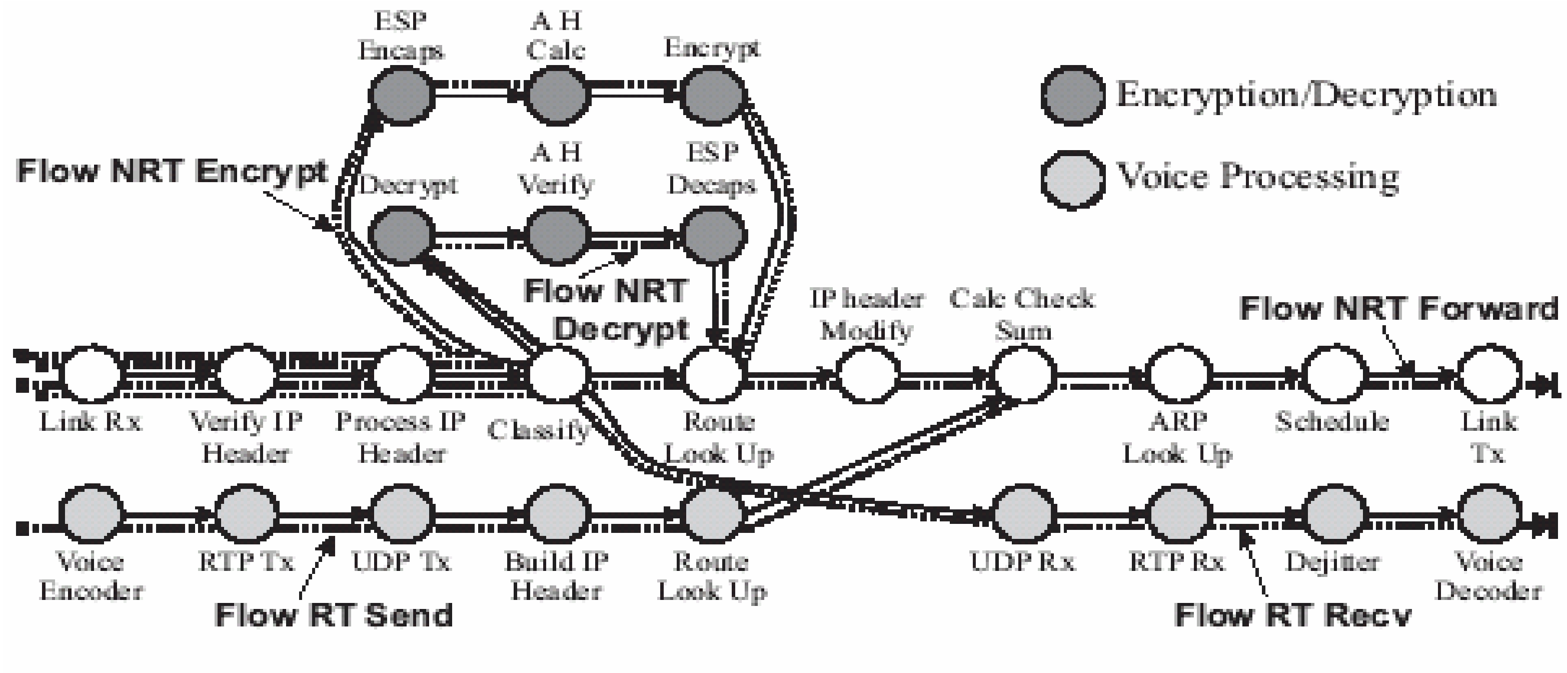
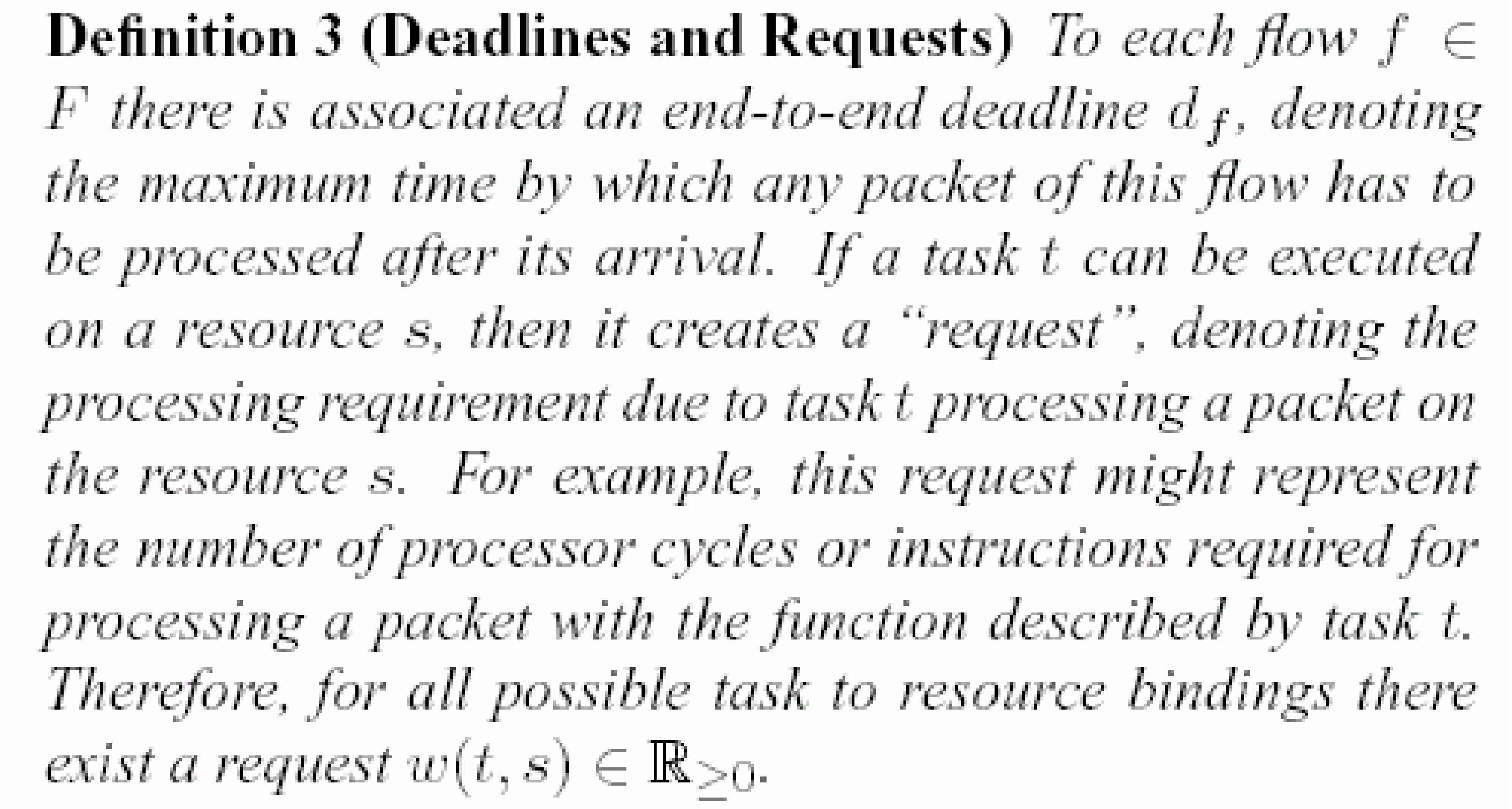


Figure 12. Task graph for a network proces-

Deadlines and Requests



Definition 3 (Deadlines and Requests) *To each flow* $f \in F$ *there is associated an end-to-end deadline* d_f, *denoting the maximum time by which any packet of this flow has to be processed after its arrival. If a task* t *can be executed on a resource* s, *then it creates a "request", denoting the processing requirement due to task* t *processing a packet on the resource* s. *For example, this request might represent the number of processor cycles or instructions required for processing a packet with the function described by task* t. *Therefore, for all possible task to resource bindings there exist a request* $w(t, s) \in \mathbb{R}_{\geq 0}$.

System Architecture

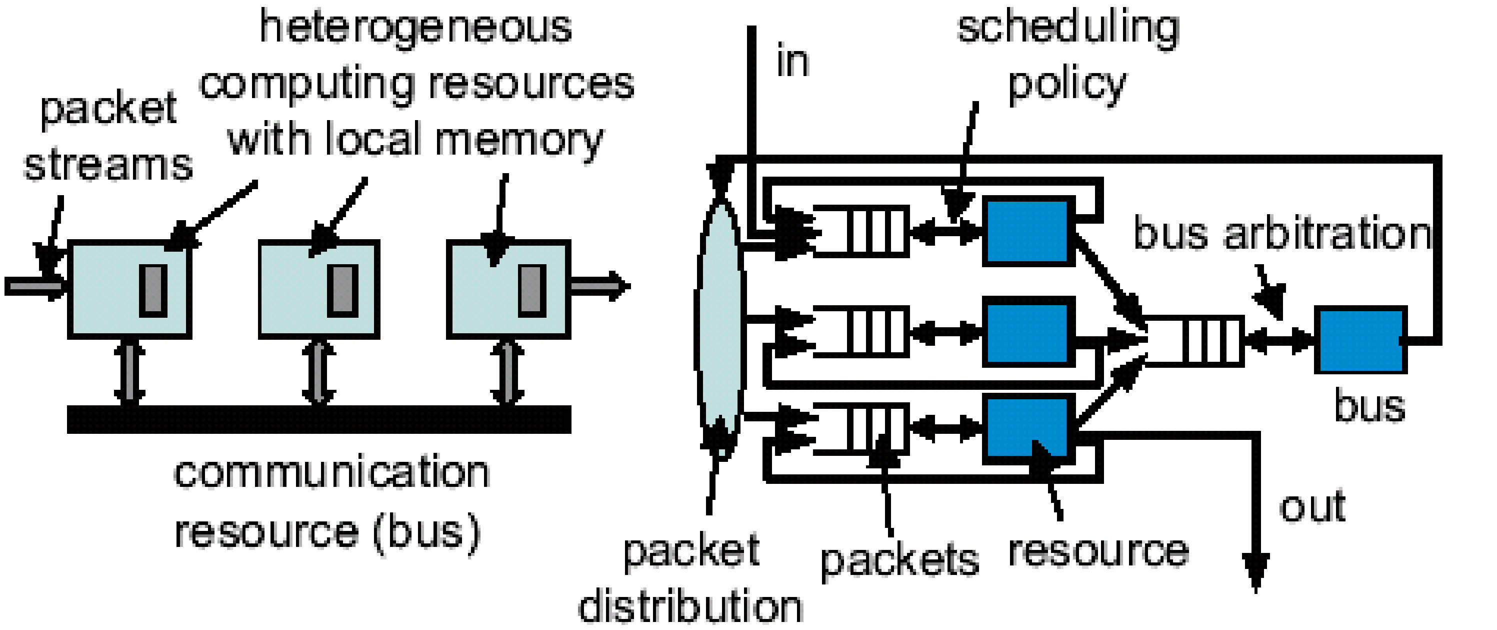
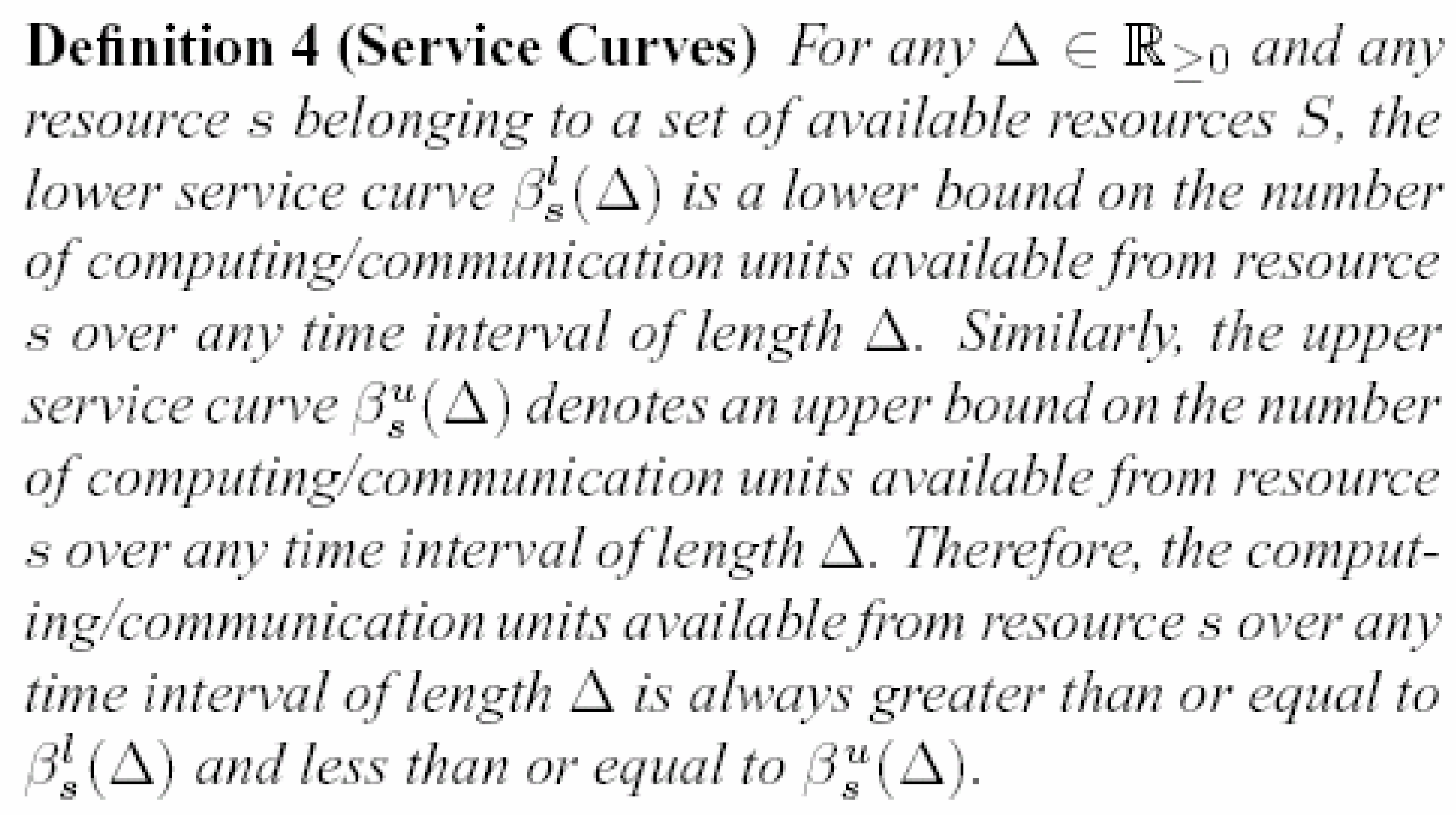


Figure 2. Example of a physical (left) and logical (right) structure of a network processor architecture.

Service Curves



Definition 4 (Service Curves) *For any $\Delta \in \mathbb{R}_{\geq 0}$ and any resource s belonging to a set of available resources S, the lower service curve $\beta_s^l(\Delta)$ is a lower bound on the number of computing/communication units available from resource s over any time interval of length Δ. Similarly, the upper service curve $\beta_s^u(\Delta)$ denotes an upper bound on the number of computing/communication units available from resource s over any time interval of length Δ. Therefore, the computing/communication units available from resource s over any time interval of length Δ is always greater than or equal to $\beta_s^l(\Delta)$ and less than or equal to $\beta_s^u(\Delta)$.*

Service Curves

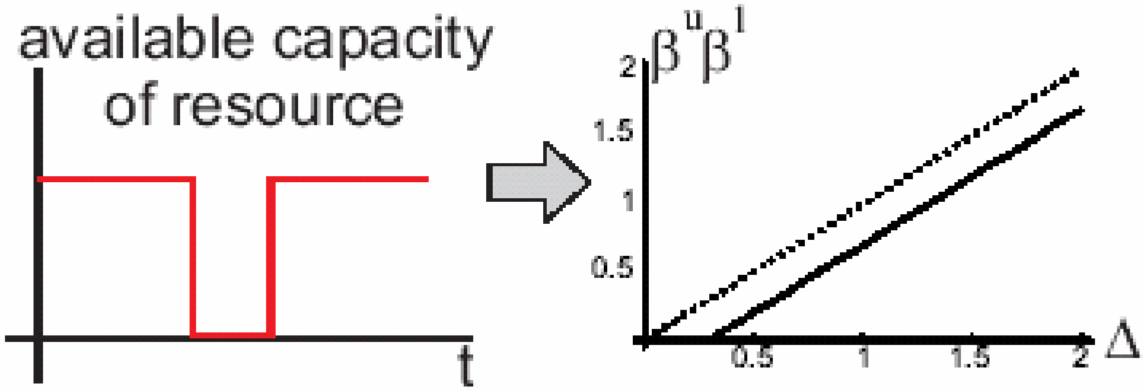
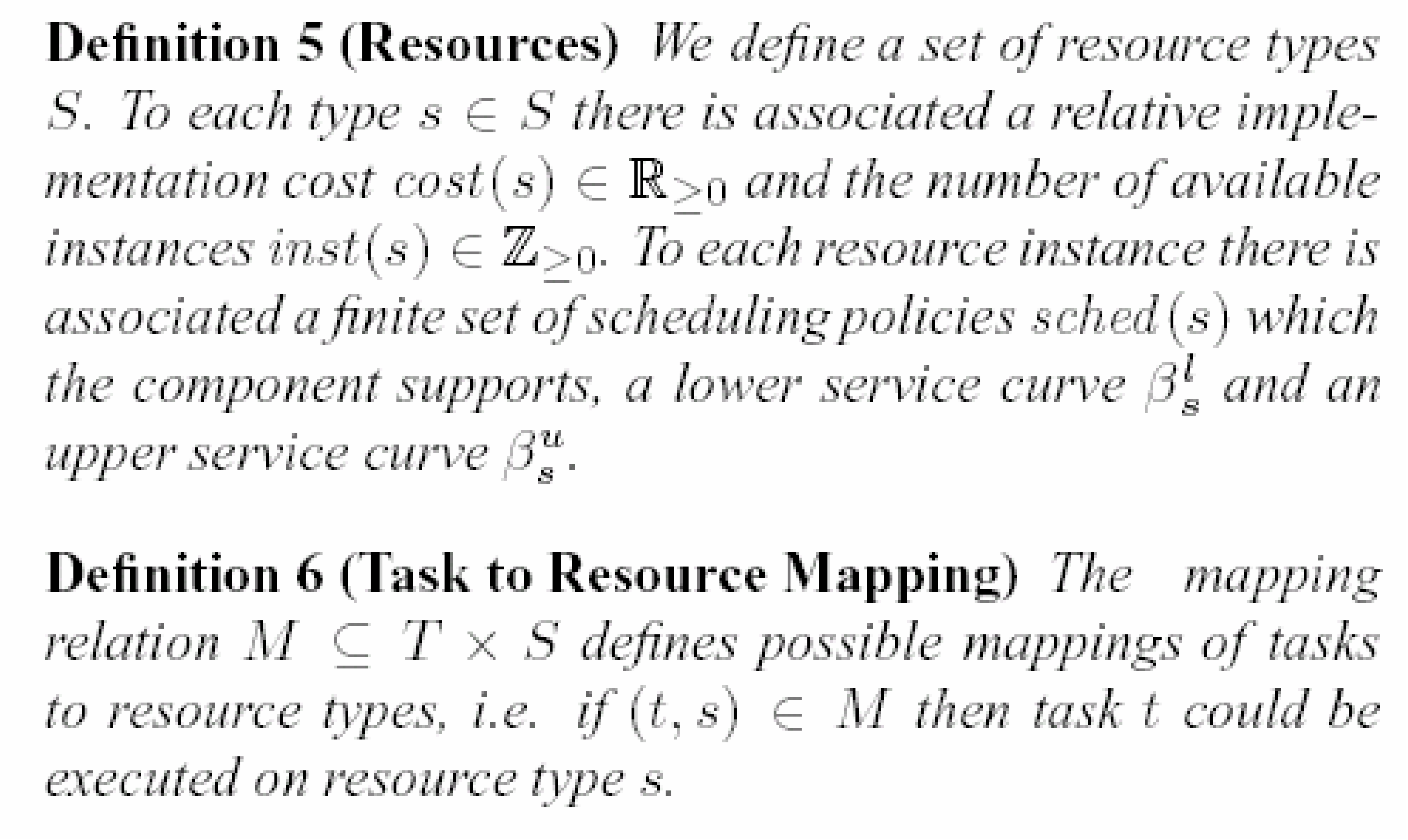


Figure 3. Representation of service curves.

Resources and Task Mapping



Definition 5 (Resources) *We define a set of resource types S. To each type $s \in S$ there is associated a relative implementation cost $cost(s) \in \mathbb{R}_{\geq 0}$ and the number of available instances $inst(s) \in \mathbb{Z}_{\geq 0}$. To each resource instance there is associated a finite set of scheduling policies $sched(s)$ which the component supports, a lower service curve β_s^l and an upper service curve β_s^u.*

Definition 6 (Task to Resource Mapping) *The mapping relation $M \subseteq T \times S$ defines possible mappings of tasks to resource types, i.e. if $(t, s) \in M$ then task t could be executed on resource type s.*

Task Mapping

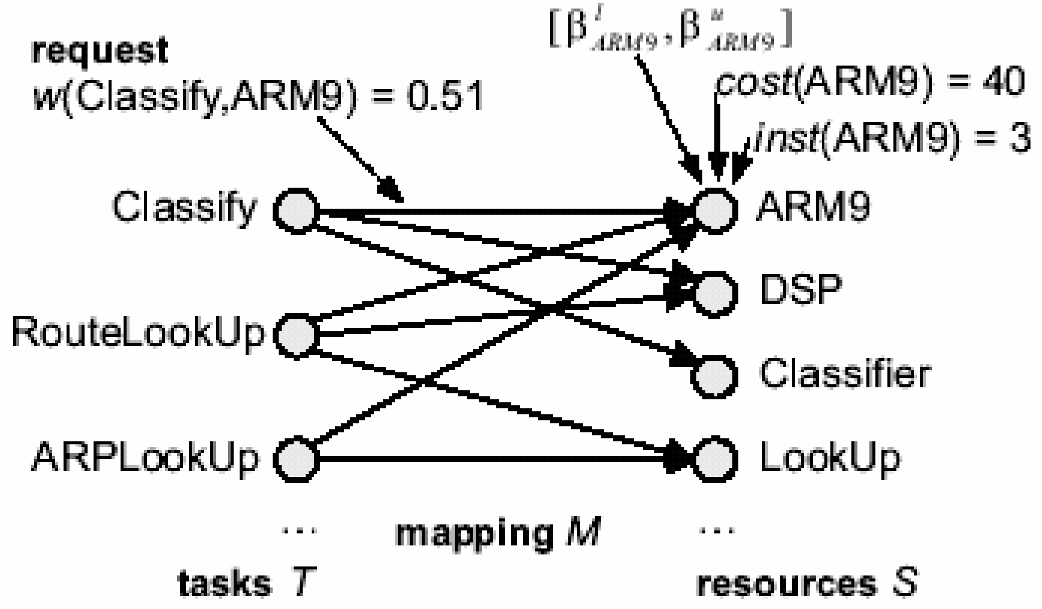
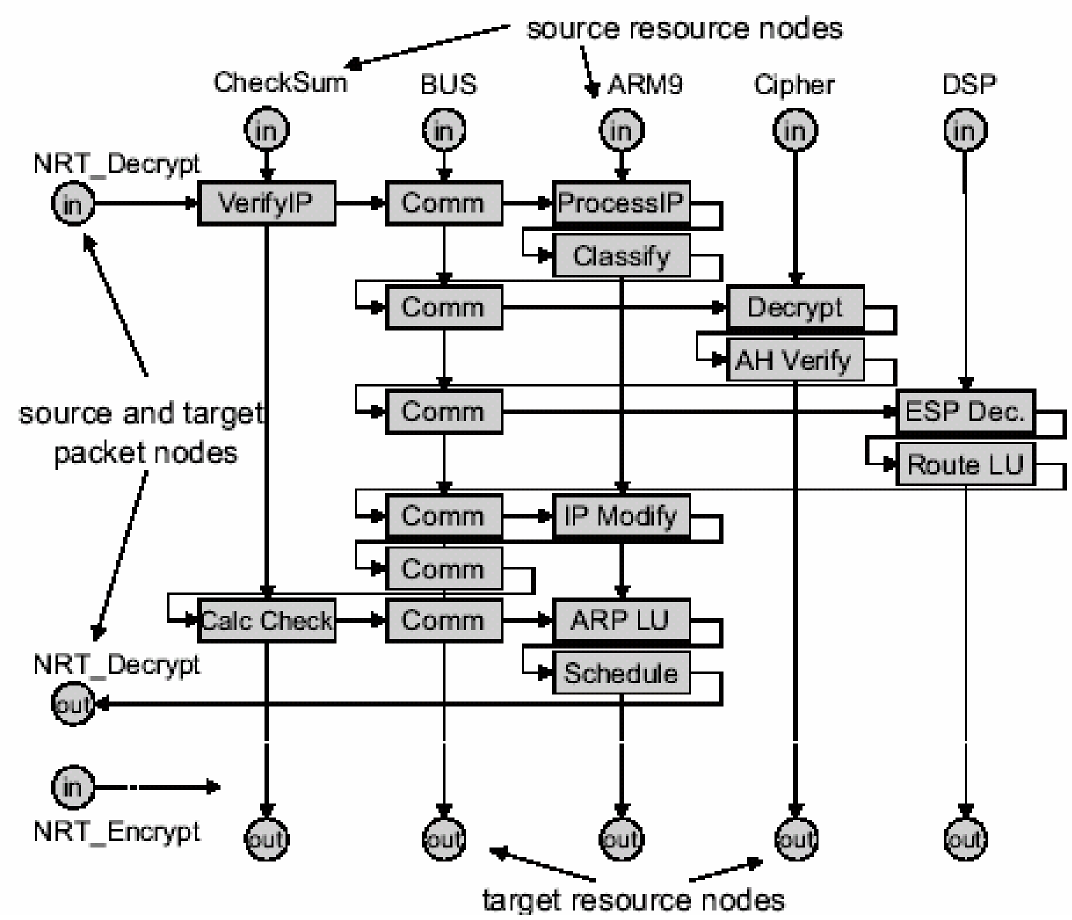


Figure 13. Graphical representation of a part of the mapping of tasks to resources.

Analysis

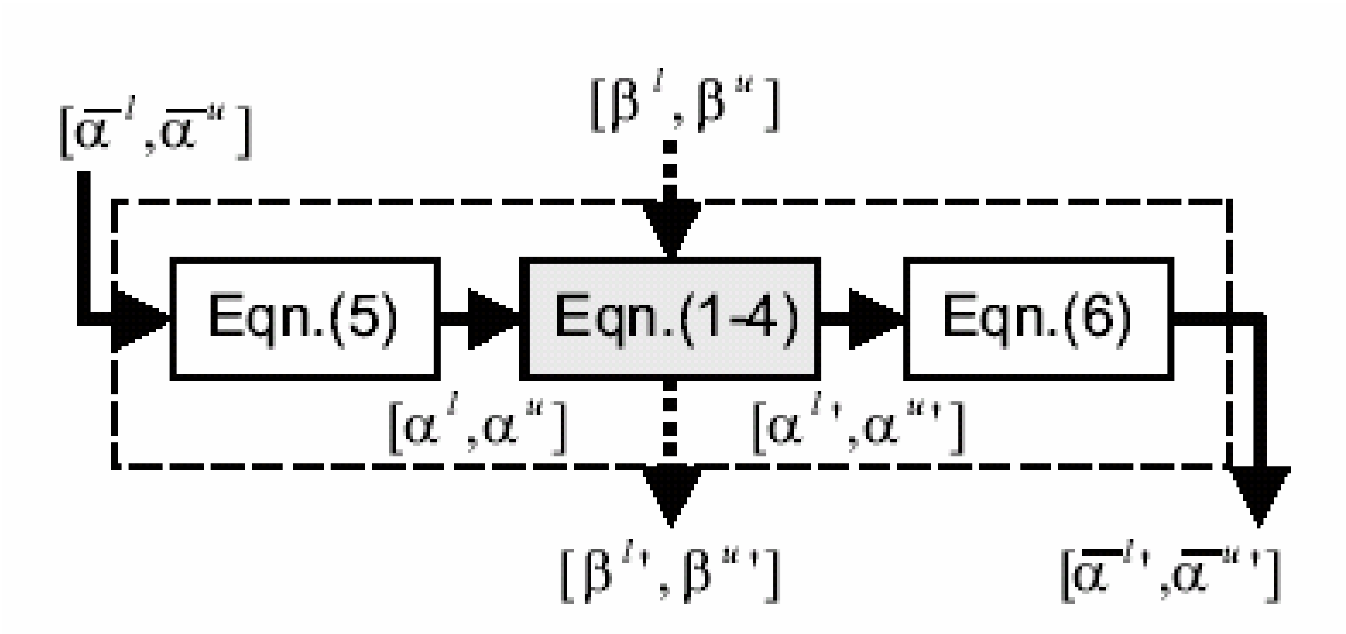
- Packets processed independently
- Characteristic chain of tasks for each flow
- Develop scheduling network
- Apply real-time calculus from node to node
- Derive bounds

Scheduling Network

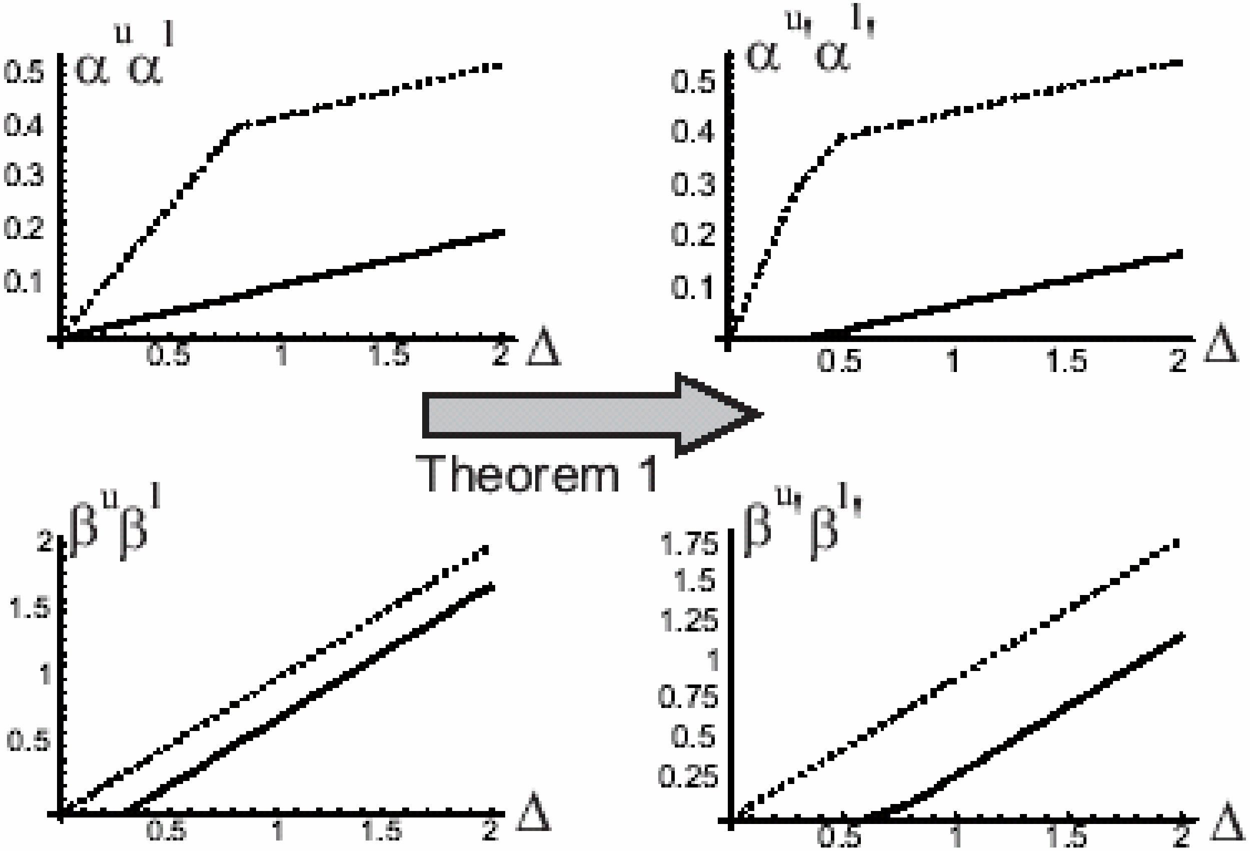


University *of* Massachusetts Amherst

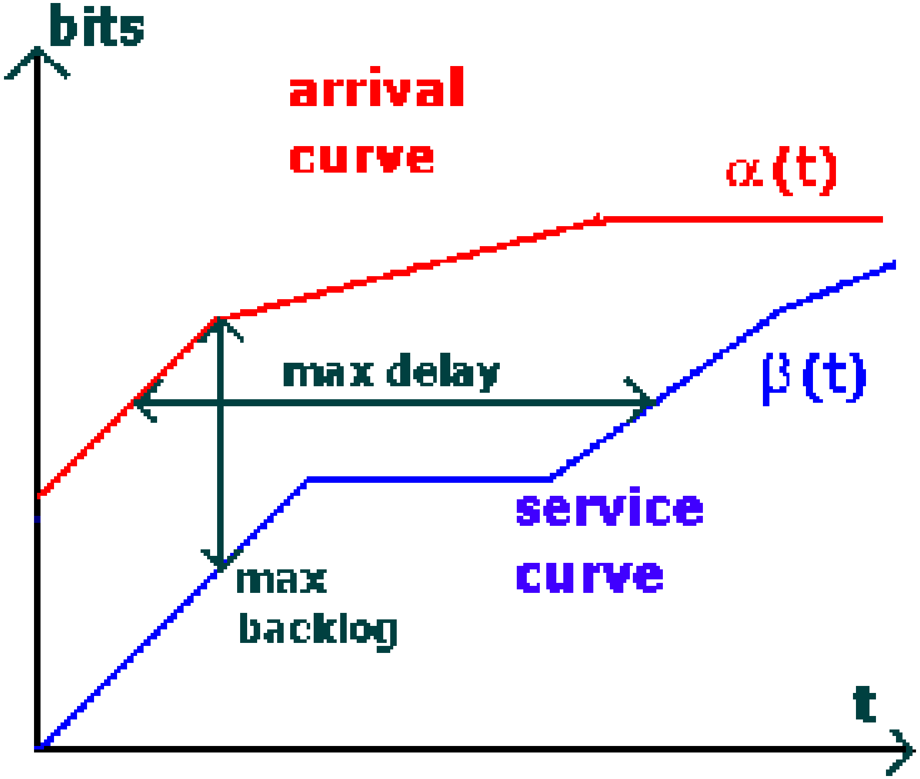
Arrival/Service Curve Transformation


$[\overline{\alpha}^{l}, \overline{\alpha}^{u}]$
$[\beta^{l}, \beta^{u}]$
Eqn.(5)
Eqn.(1-4)
Eqn.(6)
$[\alpha^{l}, \alpha^{u}]$
$[\alpha^{l\prime}, \alpha^{u\prime}]$
$[\beta^{l\prime}, \beta^{u\prime}]$
$[\overline{\alpha}^{l\prime}, \overline{\alpha}^{u\prime}]$

Arrival/Service Curve Transformation



Delay and Backlog



From: Robert Malaney & Glynn Rogers, CSIRO

Design Space Exploration

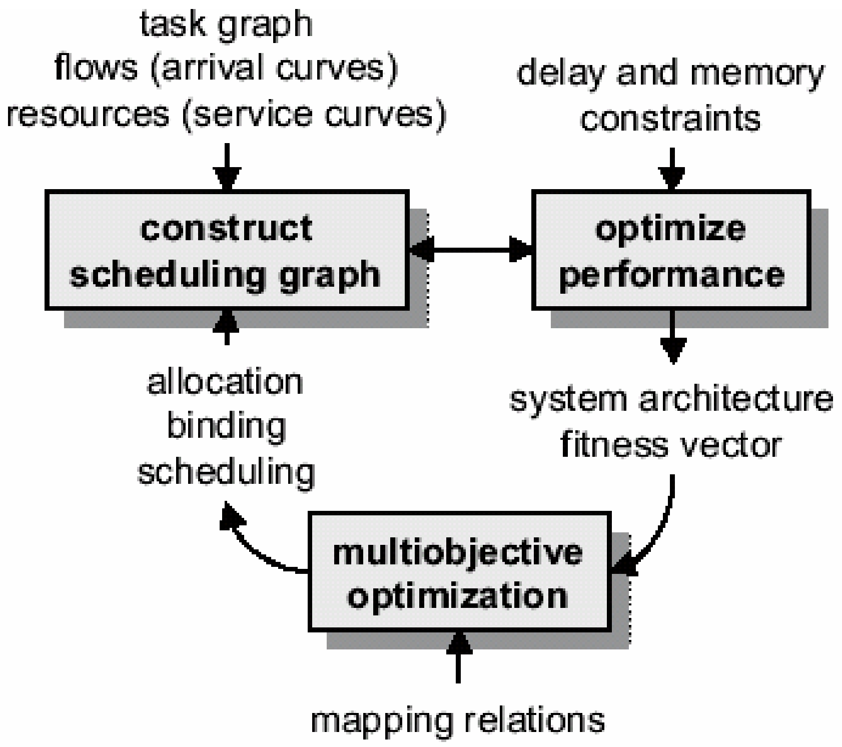
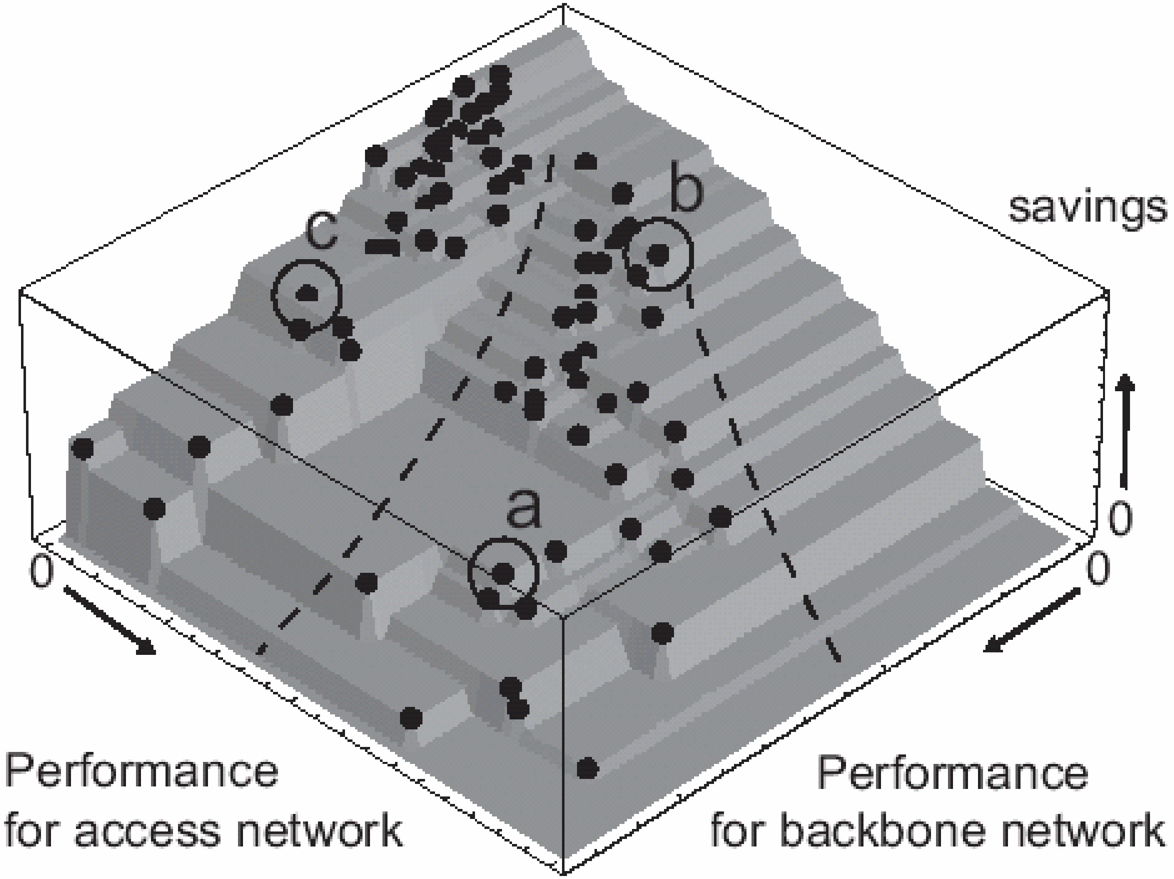


Figure 11. Basic concept for the design space exploration of packet processing systems.

Results


savings
a
b
c
0
0
0
Performance for access network
Performance for backbone network

Results

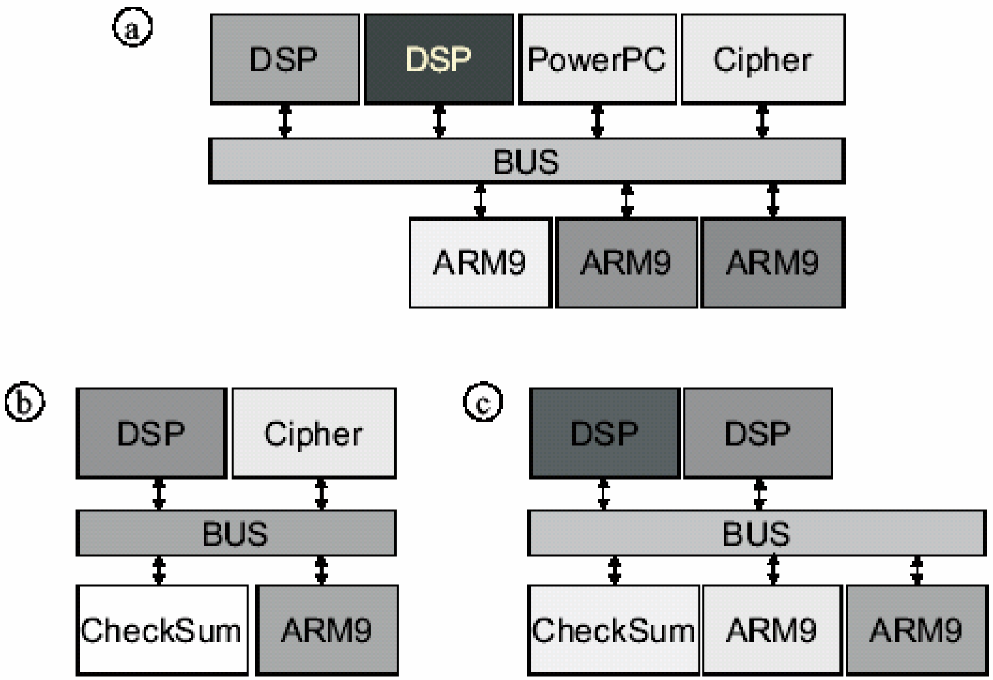


Figure 15. Examples for Pareto-optimal resource allocations taken from Fig. 14. Darker coloring means higher average utilization.

Summary

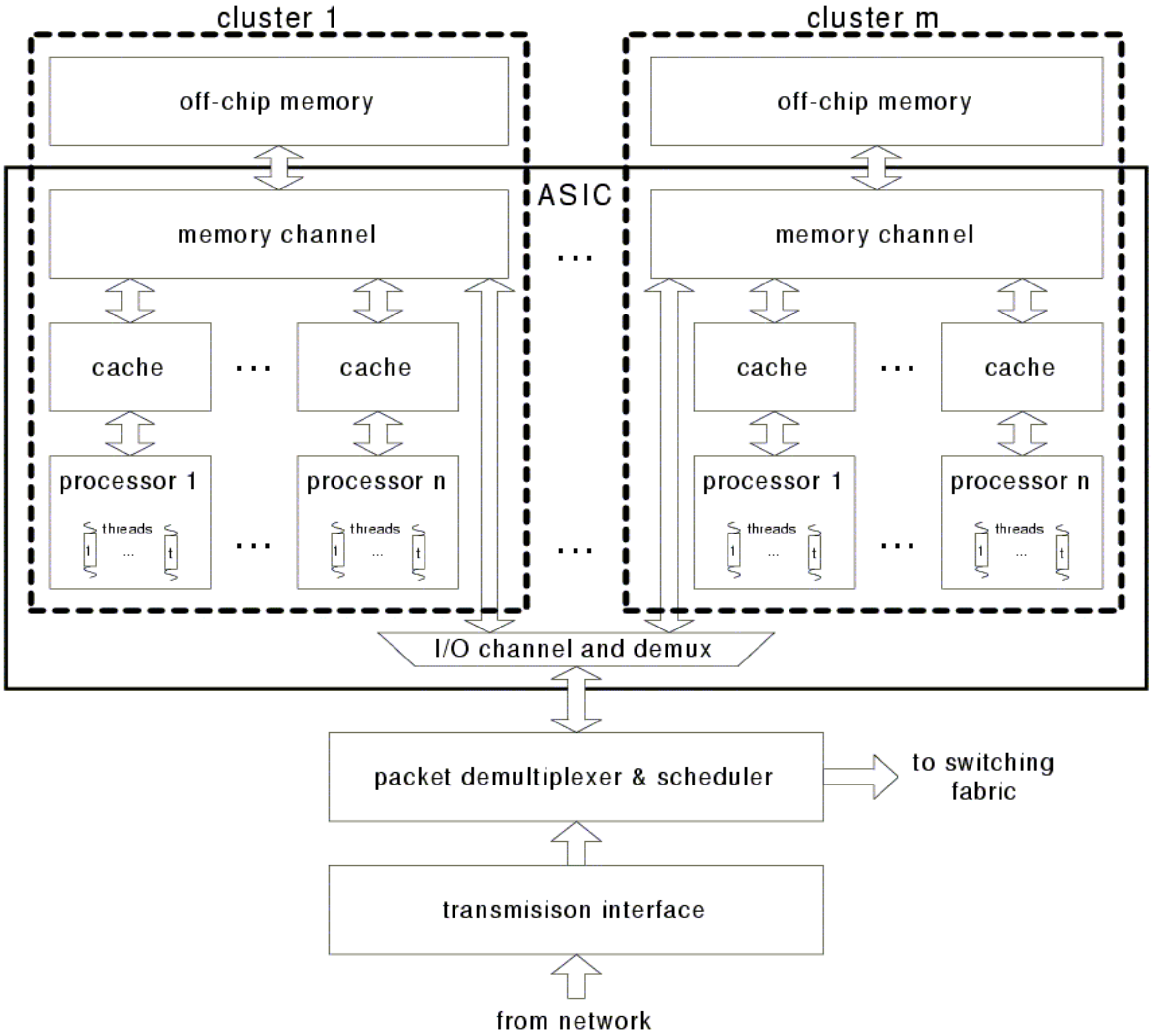
- Analytic performance model for network processors
- Models traffic, processing, and interconnect
- Service curves give upper and lower bounds
  - Might diverge over many steps
- A bit difficult to apply

Alternative Model

- Use mean-value analysis
- Compute MIPS for overall system
- Determine cost function (e.g., area)
- Parameterize with realistic workload: CommBench
- Find optimal configuration / explore design space

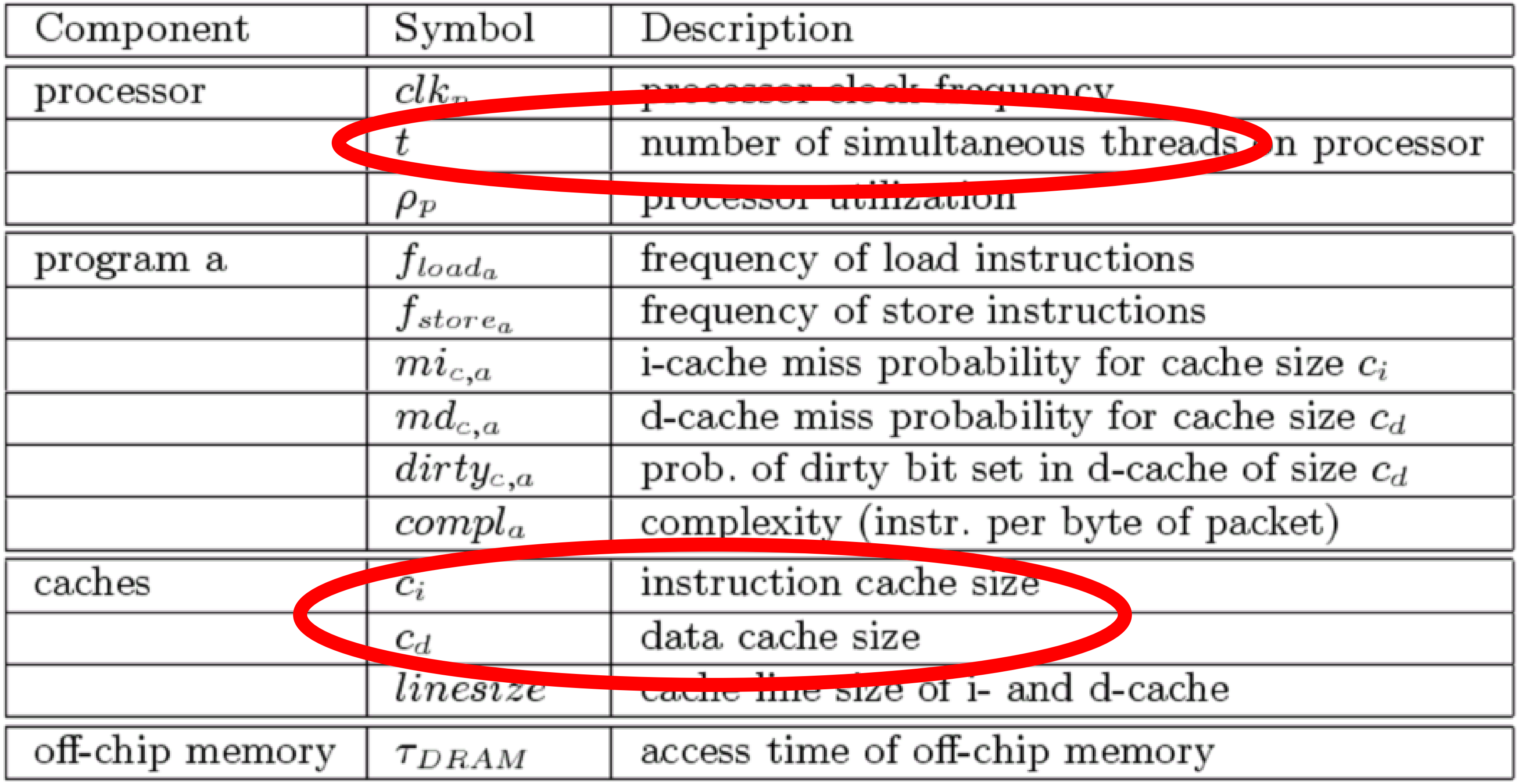
NP System Model

- Single Chip Multi-processor
- Clusters:
  - Processors
  - Per-proc cache
  - Memory channel
- Processors are simple RISC cores
- Off-chip router functions:
  - Queuing
  - Packet demux



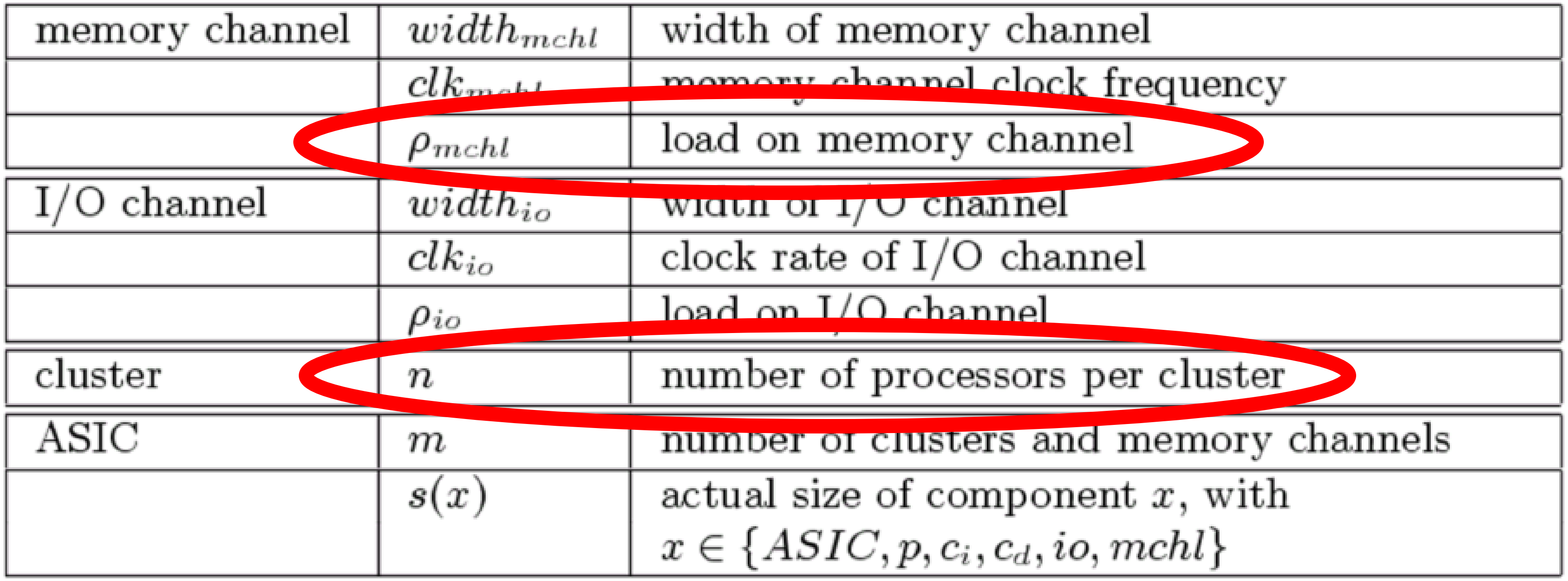
Design Parameters (1)

- Parameters that are considered in model:



Component	Symbol	Description
processor	clk_p	processor clock frequency
	t	number of simultaneous threads on processor
	ρ_p	processor utilization
program a	f_{load_a}	frequency of load instructions
	f_{store_a}	frequency of store instructions
	$mi_{c,a}$	i-cache miss probability for cache size c_i
	$md_{c,a}$	d-cache miss probability for cache size c_d
	$dirty_{c,a}$	prob. of dirty bit set in d-cache of size c_d
	$compl_a$	complexity (instr. per byte of packet)
caches	c_i	instruction cache size
	c_d	data cache size
	$linesize$	cache line size of i- and d-cache
off-chip memory	τ_{DRAM}	access time of off-chip memory

Design Parameters (2)



memory channel	$width_{mchl}$	width of memory channel
	clk_{mchl}	memory channel clock frequency
	ρ_{mchl}	load on memory channel
I/O channel	$width_{io}$	width of I/O channel
	clk_{io}	clock rate of I/O channel
	ρ_{io}	load on I/O channel
cluster	n	number of processors per cluster
ASIC	m	number of clusters and memory channels
	$s(x)$	actual size of component x, with $x \in \{ASIC, p, c_i, c_d, io, mchl\}$

- Develop performance model:
  1. Processor utilization
  2. Cache miss rate and memory access time
  3. Memory channel utilization
  4. Cluster configuration

Processing Power

- RISC: one instruction every cycle unless stalled
- Utilization ?$_p$ gives fraction of “useful” cycles
- Total processing power:

$$IPS = \sum_{j=1}^{m} \sum_{k=1}^{n} \cdot \rho_{p_{j,k}} \cdot clk_{p_{j,k}}$$

- If all processors are identical in configuration and workload:

$$IPS = m \cdot n \cdot \rho_p \cdot clk_p$$

- Question: How to determine ?$_p$?

Processor Utilization

- Cache misses cause processor stalls
  - Reduce utilization
- Multithreading hides memory access latencies
- Processor utilization [Agarwal 1992]:

$$\rho_p(t) = 1 - \frac{1}{\sum_{i=0}^{t} \left(\frac{1}{p_{miss} \cdot \tau_{mem}}\right)^{i} \frac{t!}{(t-i)!}}$$

- Utilization decreases with
  - more cache misses (p_{miss})
  - longer memory accesses (τ_{mem})
  - Fewer threads (t)
- Need to determine τ_{mem} and p_{miss}

Memory System

- Memory access time has three components:
  - Queuing time until request is served
  - DRAM access time
  - Memory line transmission time

$$\tau_{mem} = \tau_Q + \tau_{DRAM} + \tau_{transmit}$$

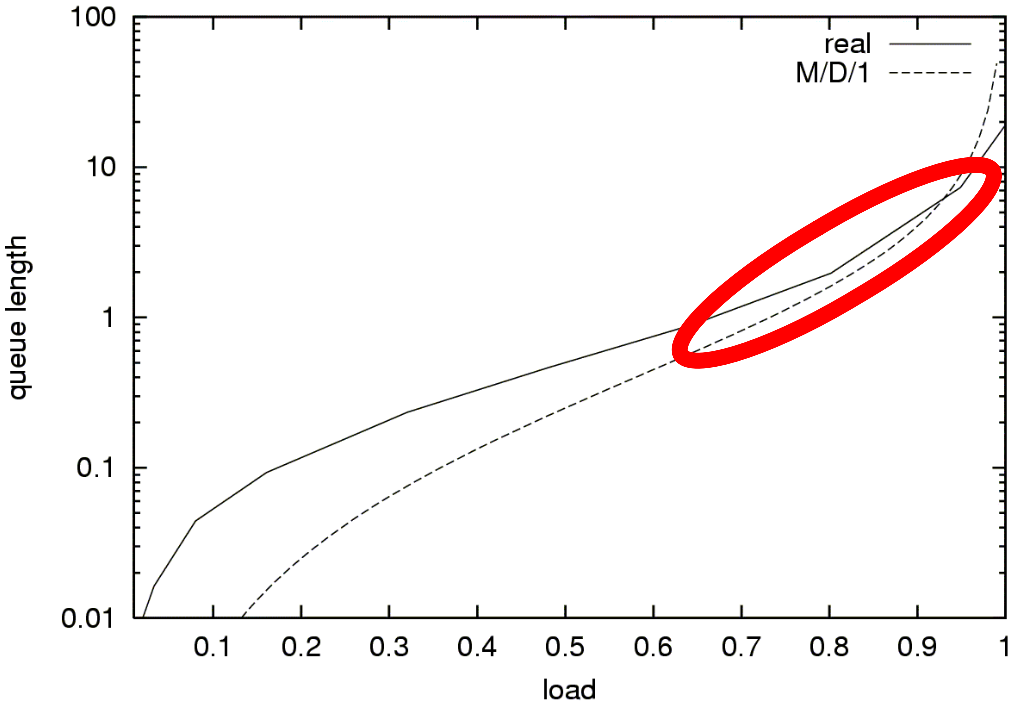
- DRAM access time fixed by technology used.
- Transmission time:

$$\tau_{transmit} = \frac{linesize}{width_{mchl}} \cdot \frac{clk_p}{clk_{mchl}}$$

- Queuing time depends on load on memory channel.

Queuing Approximation

- Processors in cluster generate memory requests
  - Single server queuing system
  - Deterministic service time
  - Geometrically distributed inter-request time
- Approximation with waiting time in M/D/1 queue:



$$\tau_Q = \frac{\rho_{mchl}^2}{2(1-\rho_{mchl})} \cdot \frac{linesize}{width_{mchl}} \cdot \frac{clk_p}{clk_{mchl}}$$

On-Chip Caches

- Miss rate is combination of i-cache and d-cache misses:

$$p_{miss,a} = mi_{c,a} + (f_{load_a} + f_{store_a}) \cdot md_{c,a}$$

- Miss rates of application depend on effective cache size.
- Threads compete for cache => cache pollution
- Cache is effectively split among threads.
  - Effective cache size:

$$c_{i,\mathit{eff}} = \frac{c_i}{t}, \quad c_{d,\mathit{eff}} = \frac{c_d}{t}$$

- We now have expression for processor utilization.

Memory and I/O Channel

- How many processor can share one memory channel?
- Processor utilization and miss rates gives memory bandwidth bw$_{\text{mchl,1}}$ of one processor.
- Number of processors that can share memory channel:

$$n = \left\lfloor \frac{width_{mchl} \cdot clk_{mchl} \cdot \rho_{mchl}}{bw_{mchl,1}} \right\rfloor$$

- Bandwidth for I/O channel depends on application:
  - Complex applications: little I/O
  - Simple applications: more I/O
  - Formal definition of "complexity" in paper
- Performance equation complete.

Chip Area

- Summation over all chip components:

$$area_{NP} = s(io) + \sum_{j=1}^{m}(s(mchl) + \sum_{k=1}^{n}(s(p_{j,k}, t) + s(c_{i_{j,k}}) + s(c_{d_{j,k}})))$$

- Processor size depends on number of thread contexts:

$$s(p,t) = s(p_{basis}) + t \cdot s(p_{thread})$$

- Memory channel size depends on channel width:

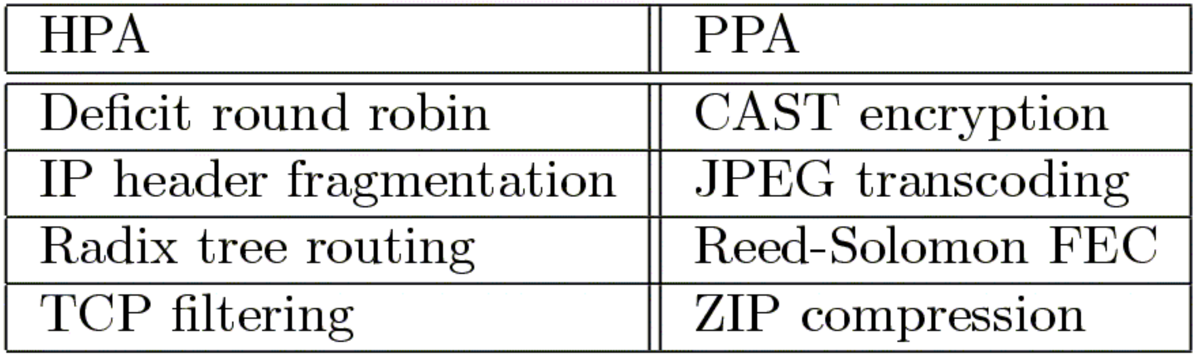
$$s(mchl) = s(mchl_{basis}) + width_{mchl} \cdot s(mchl_{pin})$$

Model Summary

- With IPS performance of system and chip area:
  - Compute IPS/area
- Necessary parameters:
  - Application parameters (load/store freq., cache miss rates, …)
  - Technology parameters (processor clock, component sizes, …)
- => Benchmark for application parameters

CommBench

- Network processor benchmark
- Benchmark applications:
  - Header-processing applications (HPA)
  - Payload-processing applications (PPA)



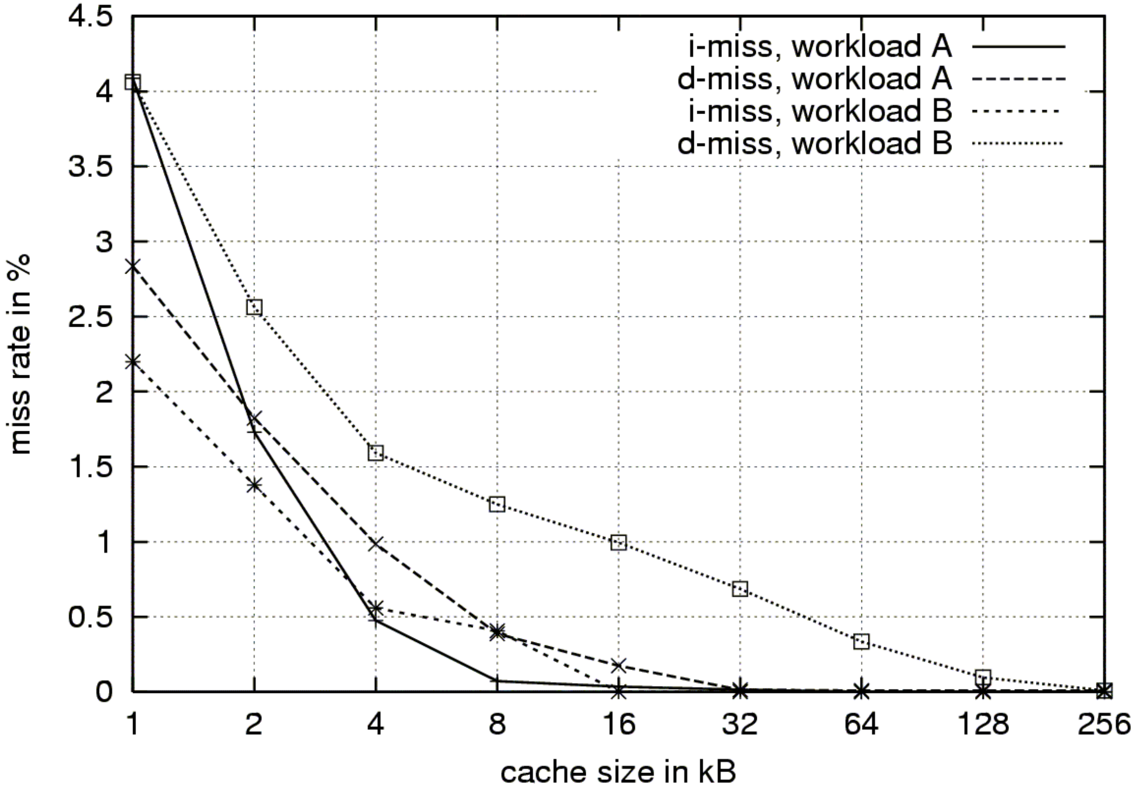
HPA	PPA
Deficit round robin	CAST encryption
IP header fragmentation	JPEG transcoding
Radix tree routing	Reed-Solomon FEC
TCP filtering	ZIP compression

- Two workloads:
  - A: HPA
  - B: PPA
- More details in [Wolf, Franklin 2000].

Application Parameters

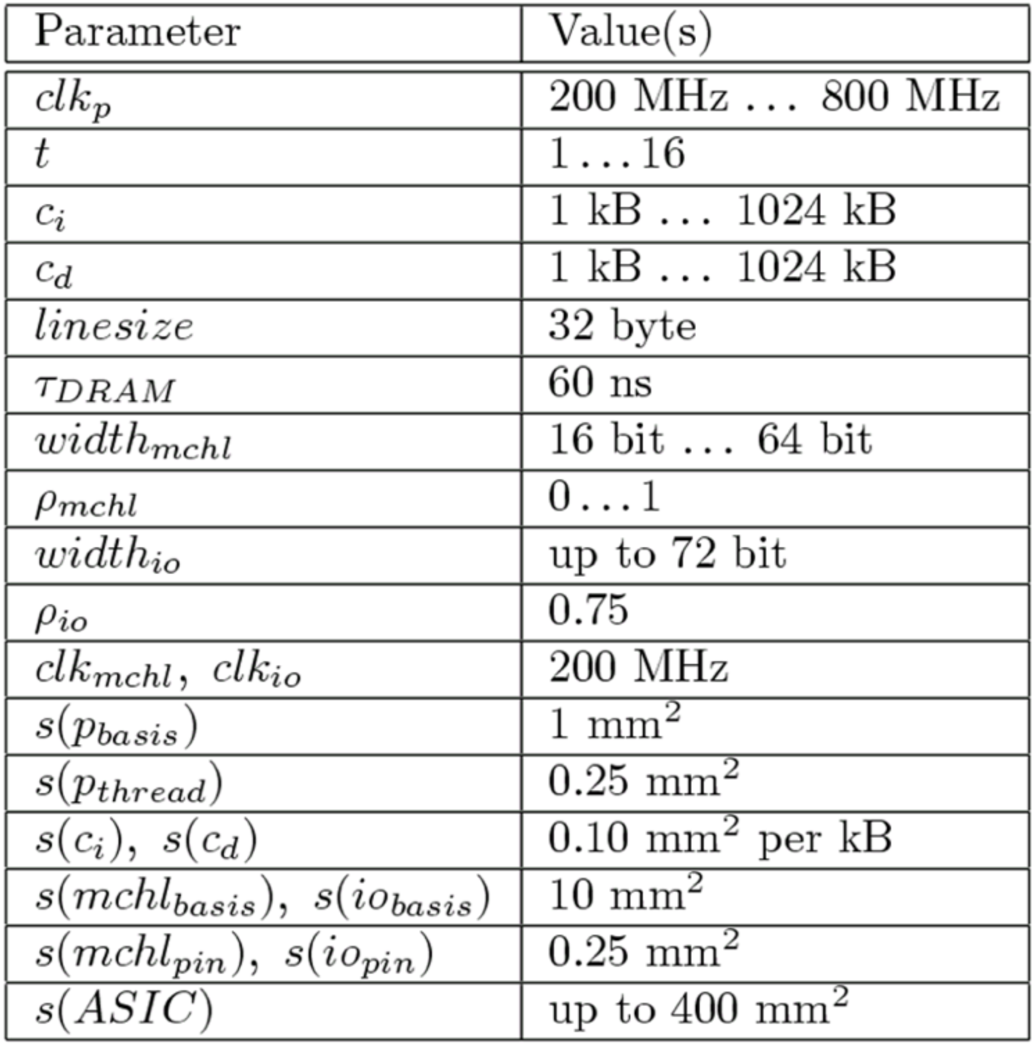
- Workload characteristics for model evaluation
- Simple parameters
  - Can easily be measured
  - Easily adaptable to other workloads

Workload W	$f_{load,W}$	$f_{store,W}$
A - HPA	0.2319	0.0650
B - PPA	0.1691	0.0595



Technology Parameters

- 0.18 µm CMOS technology
- Exact values are hard to get from industrial sources
  - Performance model also works with more accurate parameters
- Varied parameters:
  - Processor clock
  - # of threads
  - Cache sizes
  - Memory channel bandwidth and load



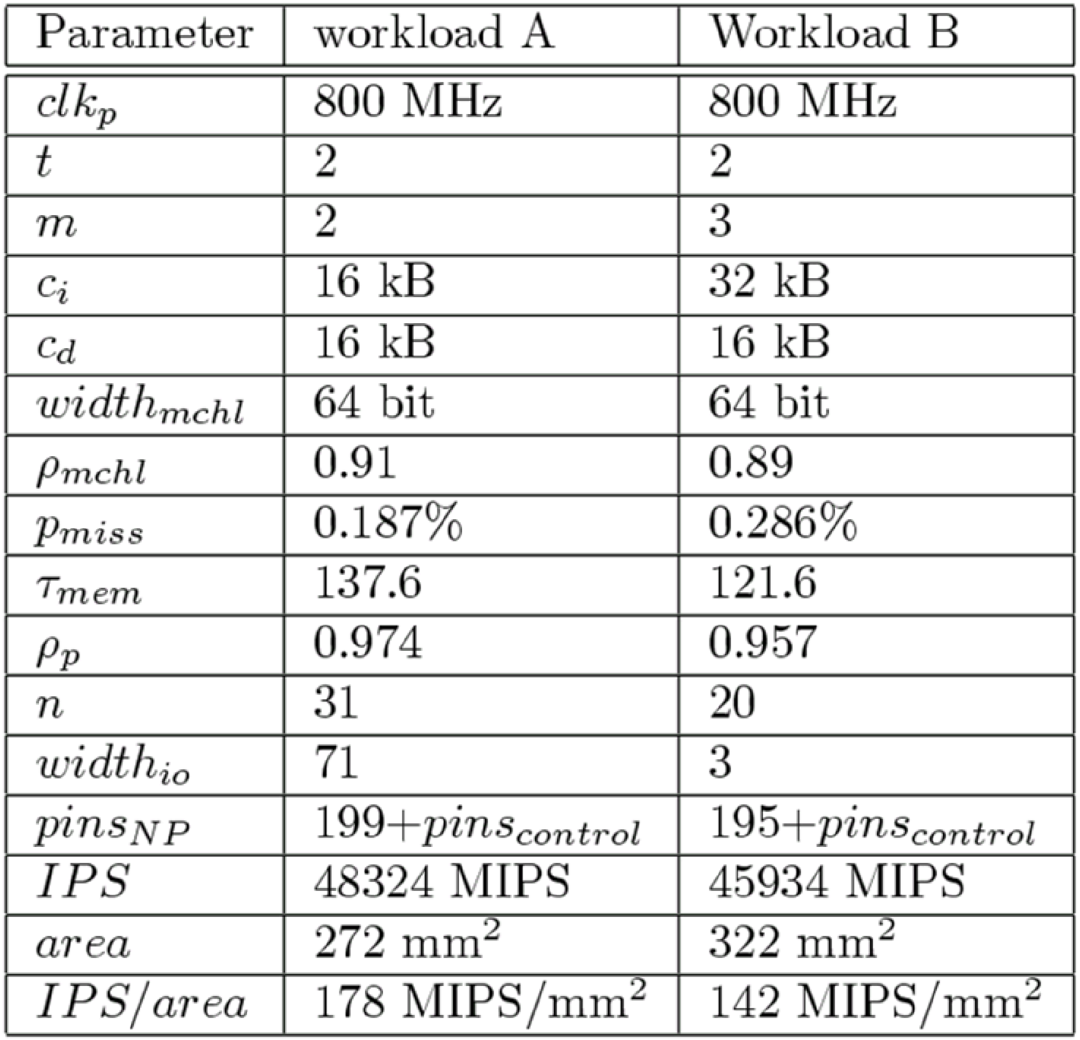
Parameter	Value(s)
clk_p	200 MHz ... 800 MHz
t	$1 \ldots 16$
c_i	1 kB ... 1024 kB
c_d	1 kB ... 1024 kB
$linesize$	32 byte
τ_{DRAM}	60 ns
$width_{mchl}$	16 bit ... 64 bit
ρ_{mchl}	$0 \ldots 1$
$width_{io}$	up to 72 bit
ρ_{io}	0.75
clk_{mchl}, clk_{io}	200 MHz
$s(p_{basis})$	1 mm^2
$s(p_{thread})$	0.25 mm^2
$s(c_i)$, $s(c_d)$	0.10 mm^2 per kB
$s(mchl_{basis})$, $s(io_{basis})$	10 mm^2
$s(mchl_{pin})$, $s(io_{pin})$	0.25 mm^2
$s(ASIC)$	up to 400 mm^2

Results

- Optimal configurations
- Performance trends (take optimal configuration and vary parameter)
  - Memory channel
  - Processor clock and threads
  - Caches
- Note: performance metric is MIPS/mm^2

Optimal Configuration

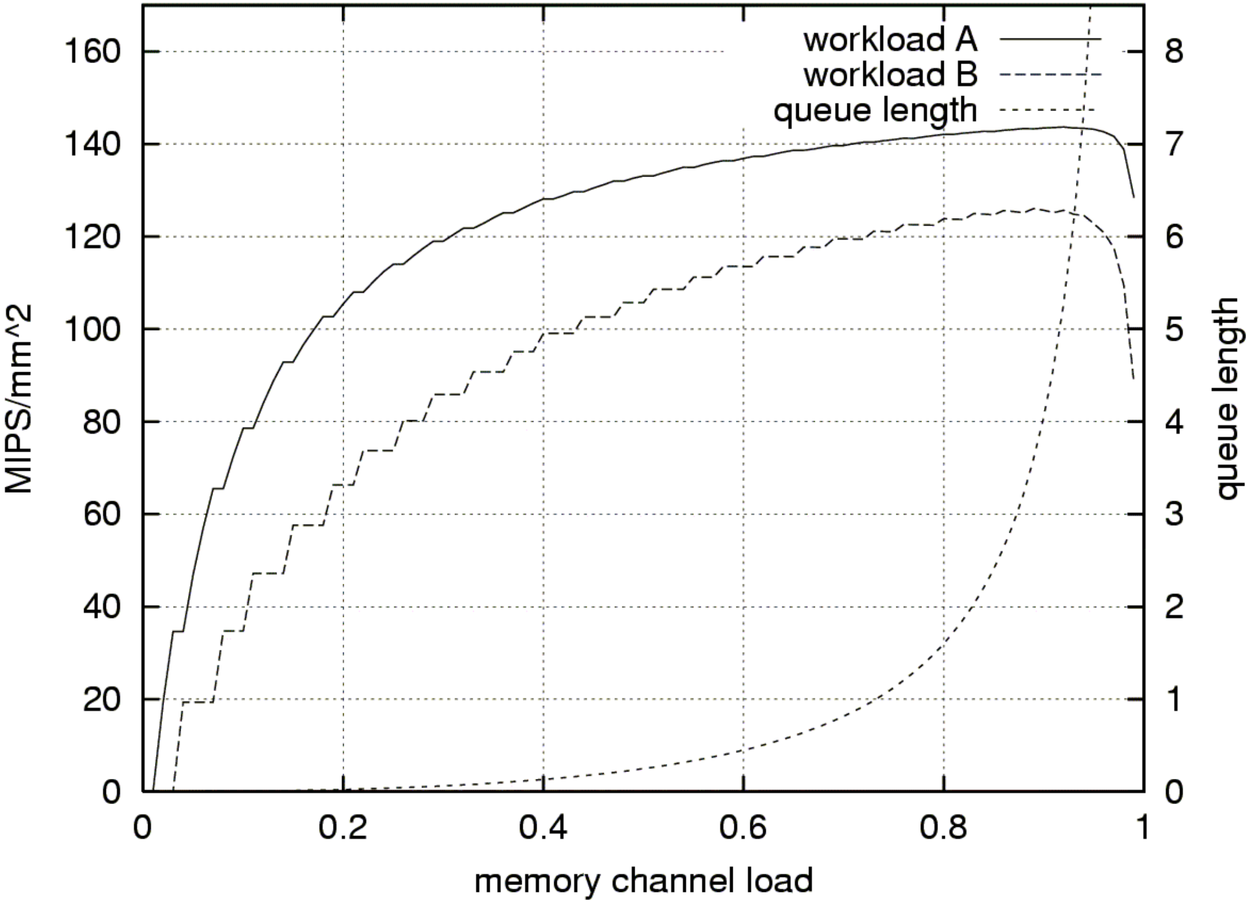
- Processor:
  - 800 MHz
  - 2 threads
  - ~96% utilization
- Chip configuration:
  - 2-3 clusters with 20-30 processors
  - 16-32 kB caches (instruction and data)
- Memory channel:
  - ~90% load
  - 64 bit width
- Off-chip memory:
  - 120-140 cycles access time
- Area:
  - 140-180 mm^2



Parameter	workload A	Workload B
clk_p	800 MHz	800 MHz
t	2	2
m	2	3
c_i	16 kB	32 kB
c_d	16 kB	16 kB
$width_{mchl}$	64 bit	64 bit
ρ_{mchl}	0.91	0.89
p_{miss}	0.187%	0.286%
τ_{mem}	137.6	121.6
ρ_p	0.974	0.957
n	31	20
$width_{io}$	71	3
$pins_{NP}$	199+$pins_{control}$	195+$pins_{control}$
IPS	48324 MIPS	45934 MIPS
$area$	272 mm^2	322 mm^2
$IPS/area$	178 MIPS/mm^2	142 MIPS/mm^2

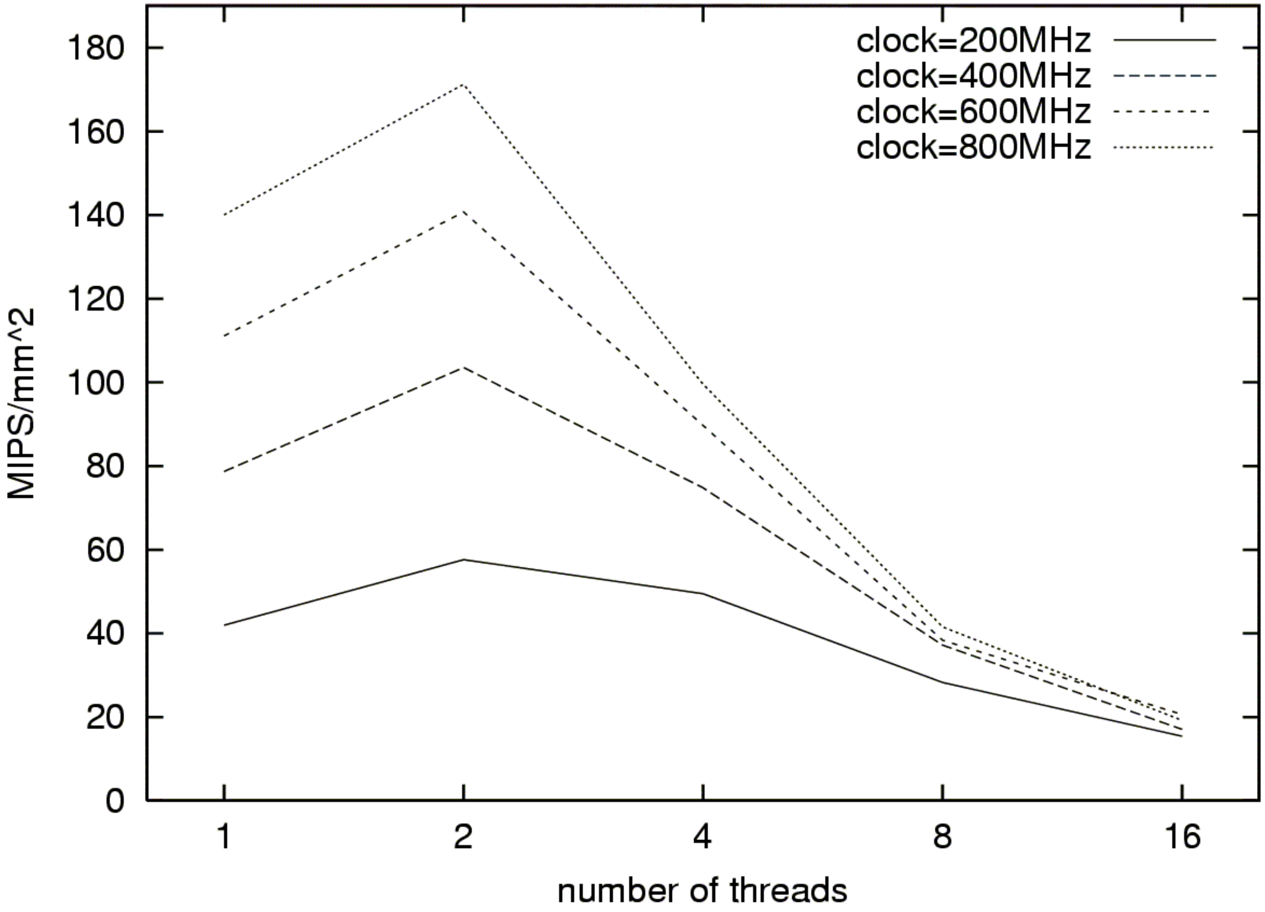
Memory Channel

- Best load ~90%
- Low load:
  - Waste of area for memory channel
- High load:
  - Very long queue length in M/D/1 model
  - High memory access time



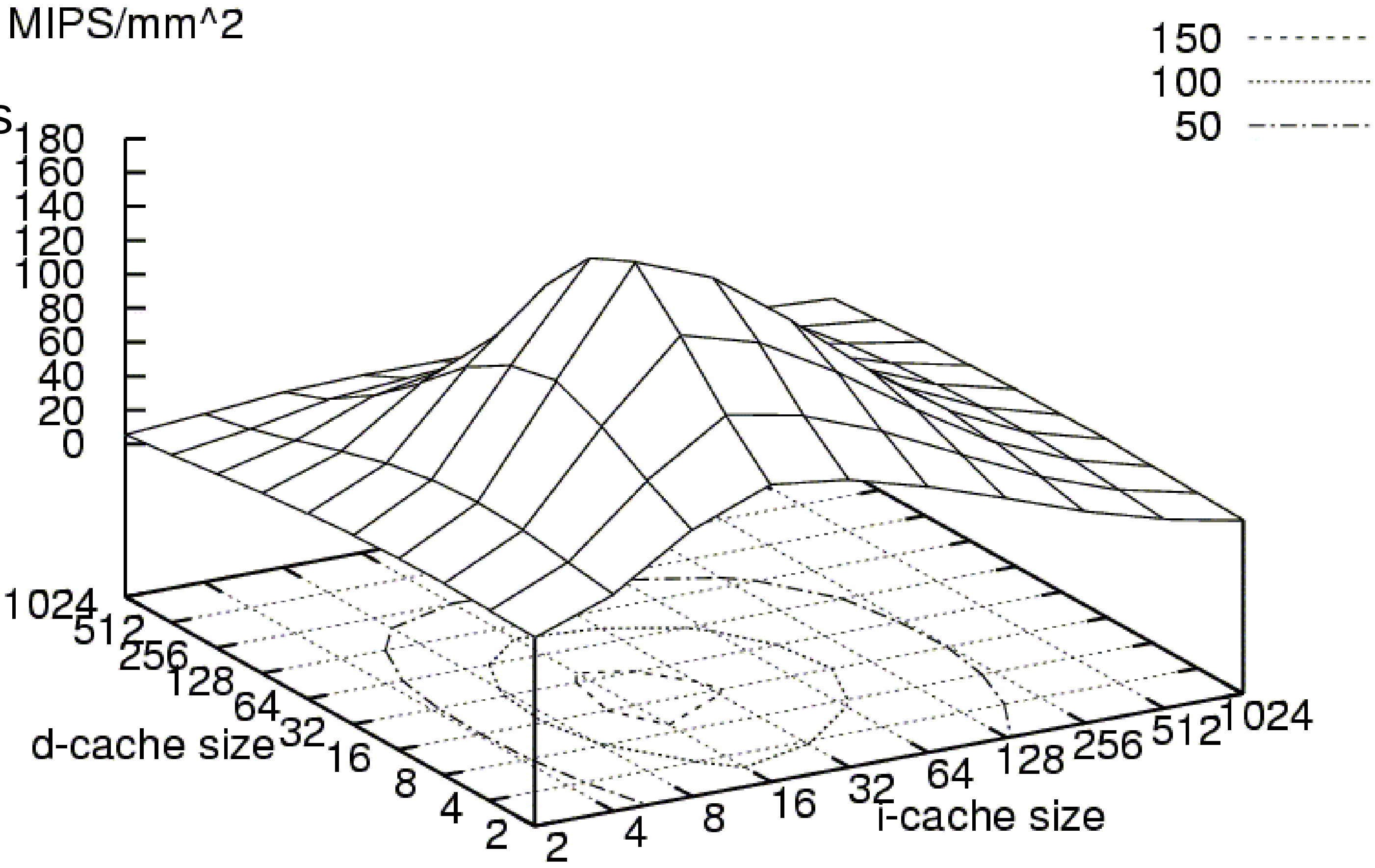
Processor Clock

- Practically linear growth with processor clock speed
- Less significant with more threads
  - Less cache per thread
  - Performance limited by off-chip memory access time



Cache Configuration

- Cache size
  - Small caches cause inefficient execution
  - Large caches waste space
- Performance very sensitive to deviations from optimum



Summary

- NP performance model
  - Determines processing performance of NP configuration
  - Relates processing power to area of system-on-a-chip
  - Uses simple workload characteristics and technology parameters
- Optimal configuration for given scenario
- Performance trends as “rules of thumb:”
  - Cache configuration has big impact on performance
  - Two to four thread contexts is optimal
  - Higher processor clock rates and memory channel directly translate into higher performance
- Model can aid in first-order NP design

Projects

- Due: 12/10/02
- Turn in:
  - Report (about 10 pages)
  - Basically conference paper style
- In class:
  - 10-12 minute presentation
  - Brief repetition of problem
  - Focus on results

Final

- December 16th, 10:30am-12:00pm, MRST 220
- Comprehensive
- Similar to midterm