Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG
Institute of Computer Science
Chair of Communication Networks
Prof. Dr.-Ing. P. Tran-Gia

Performance Modeling of Softwarized Network Functions using Discrete-Time Analysis*

Steffen Gebert, Thomas Zinner, Stanislav Lange,
Christian Schwartz, Phuoc Tran-Gia

comnet.informatik.uni-wuerzburg.de

Fachgruppentatung ITG 5.2.1, 31.08.2016

Technische Universität Hamburg-Harburg

SENDATE
PLANETS

Original Paper

Gebert, S., Zinner, T., Lange, S., Schwartz, C., Tran-Gia, P.:
Performance Modeling of Softwarized Network Functions Using Discrete-Time Analysis.
28th International Teletraffic Congress (ITC), Würzburg, Germany (2016).

Available: http://www.comnet.informatik.uni-wuerzburg.de/staff/members/steffen_gebert/

SENDATE PLANETS

http://www.sendate.eu/sendate-planets/

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

Motivation

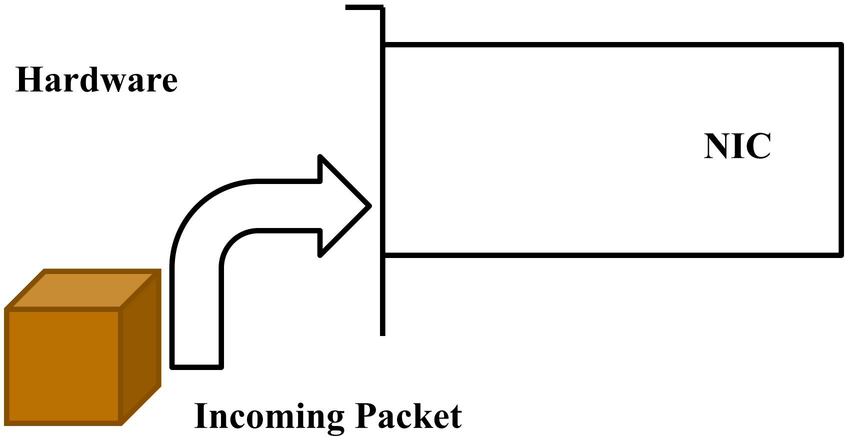
- NFV moves network functions from ASICs to server hardware
  - “optimized devices” vs. ”standard devices”
  - What are the performance implications?
- Performance optimization techniques
  - Tuning of networking stack
  - Tuning of network interface cards (NIC)
  - Packet processing frameworks (netmap, DPDK and friends)
  - Hardware offloading
- How can we estimate (good and bad) effects on network traffic?
  - Measurement / benchmarking
  - Analytical modeling

Packet Processing in an x86 System

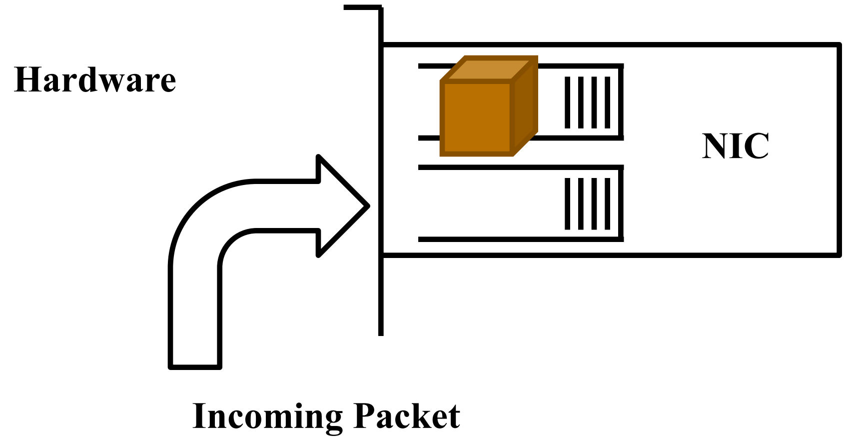
LIFE OF A PACKET

Steffen Gebert, *Thomas Zinner*

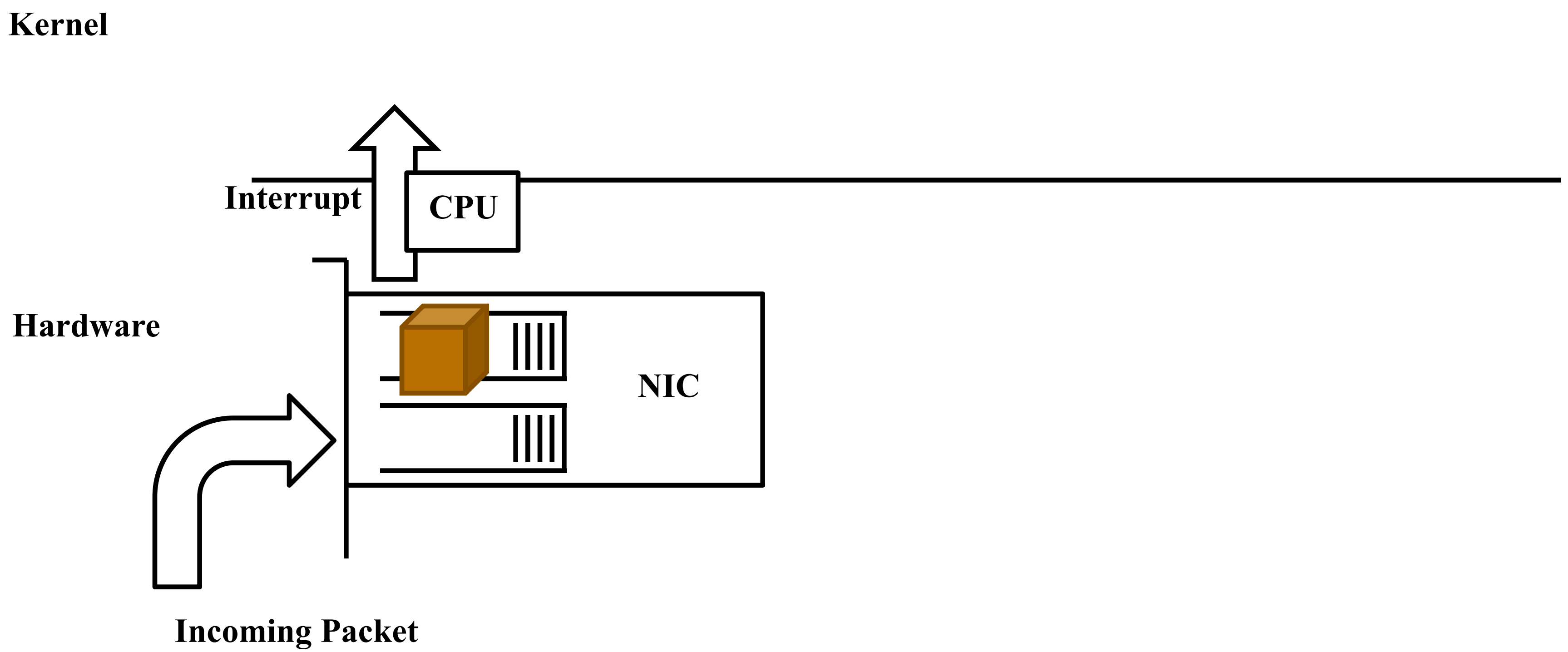
“Life of a Packet”



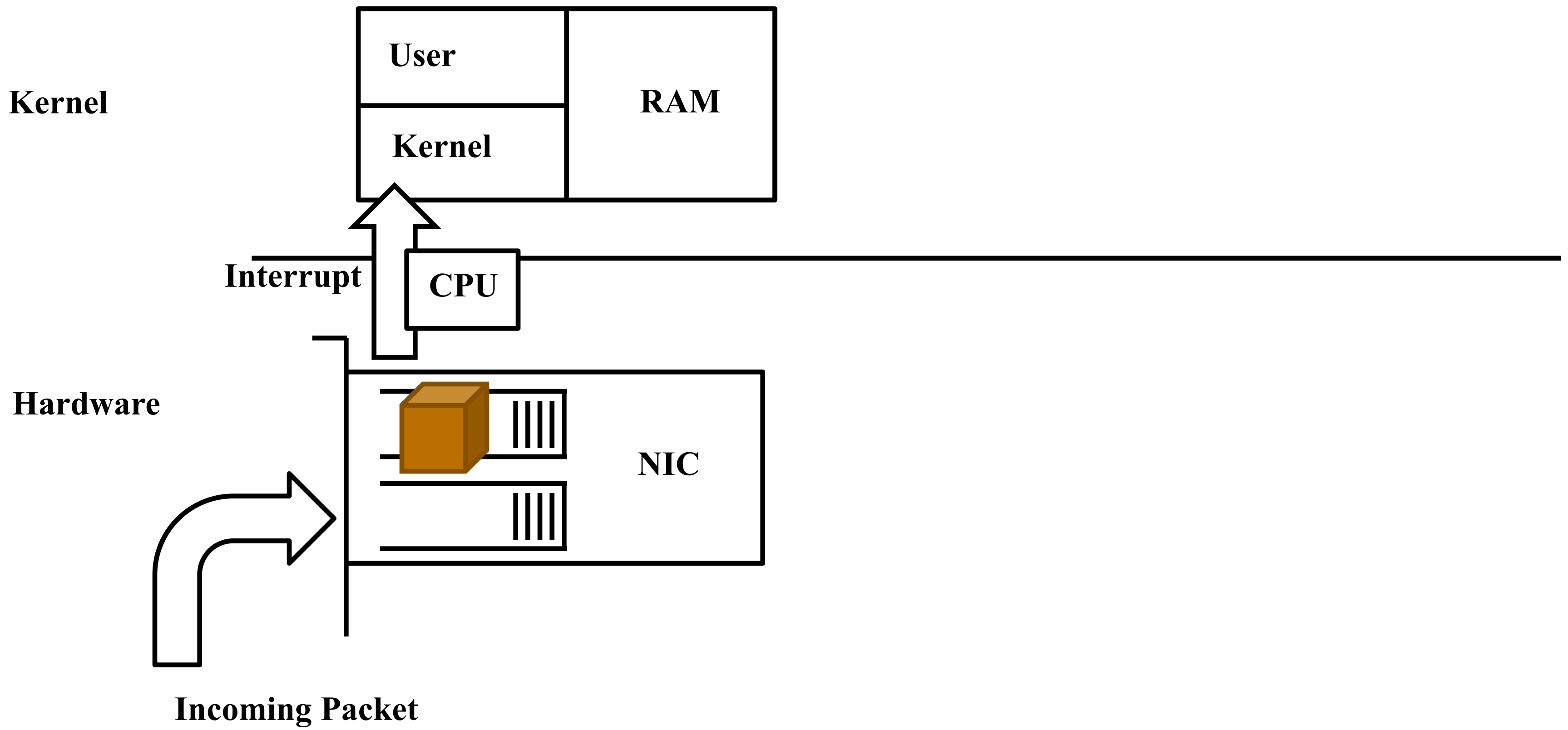
“Life of a Packet”



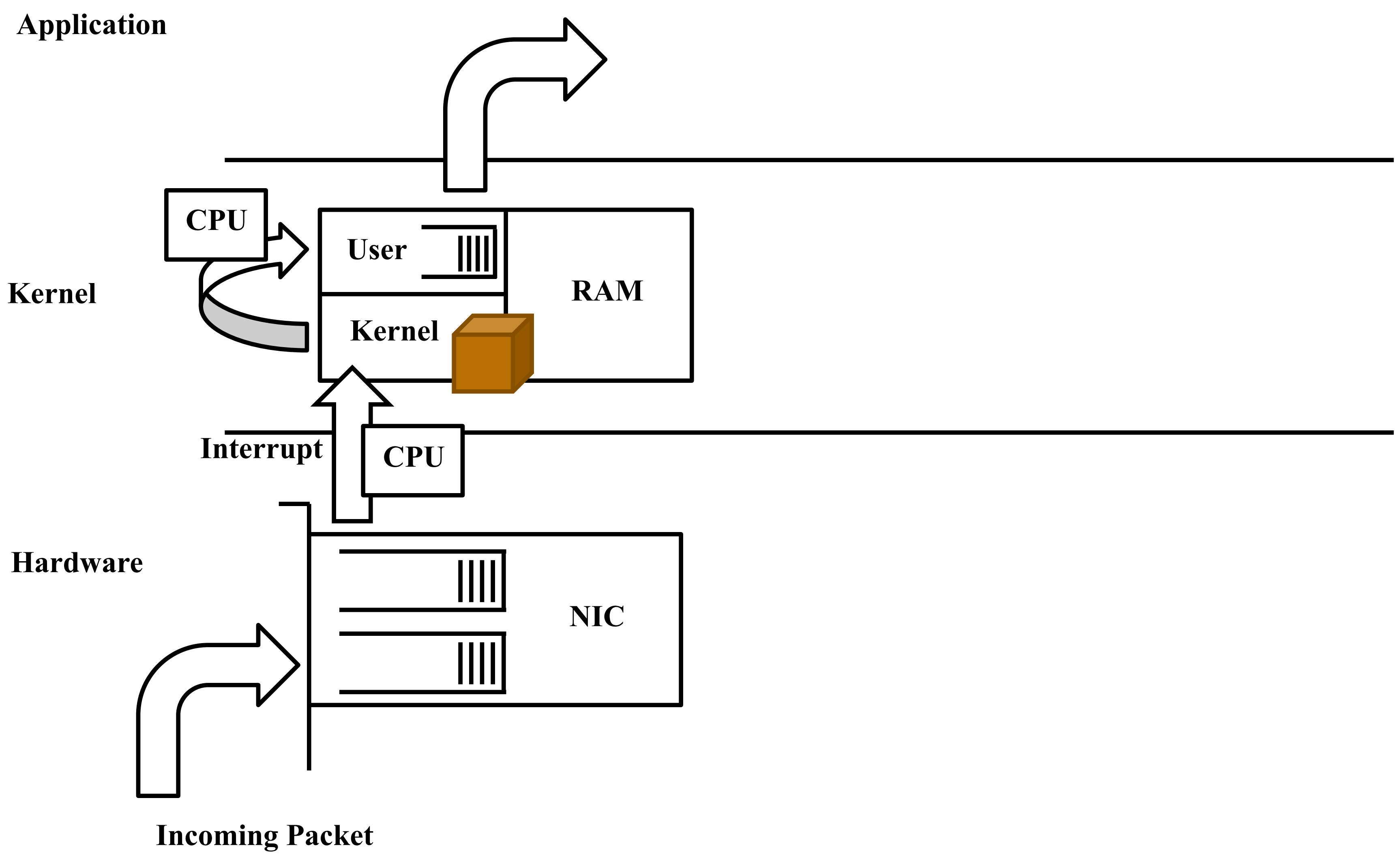
"Life of a Packet"



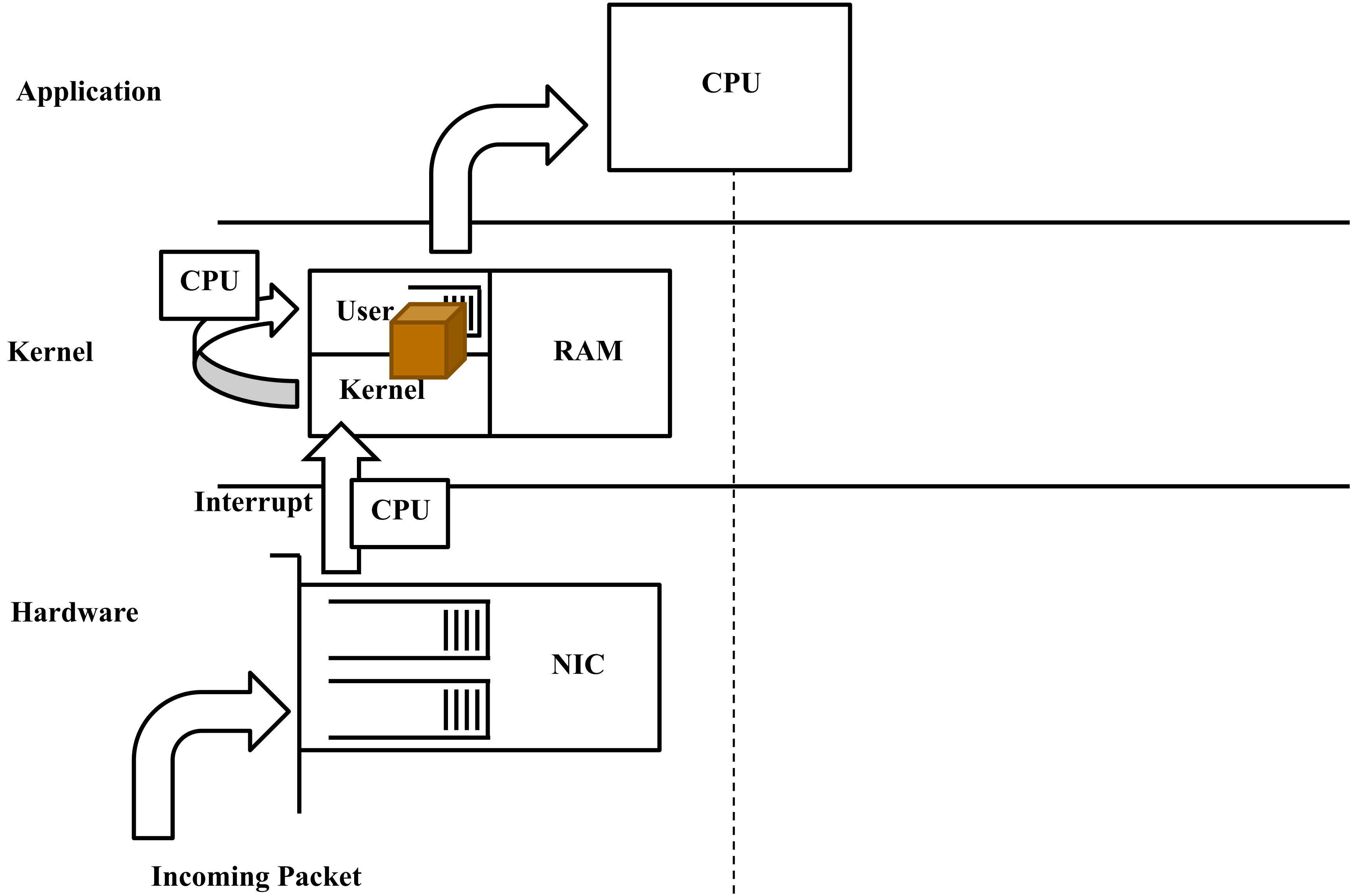
"Life of a Packet"



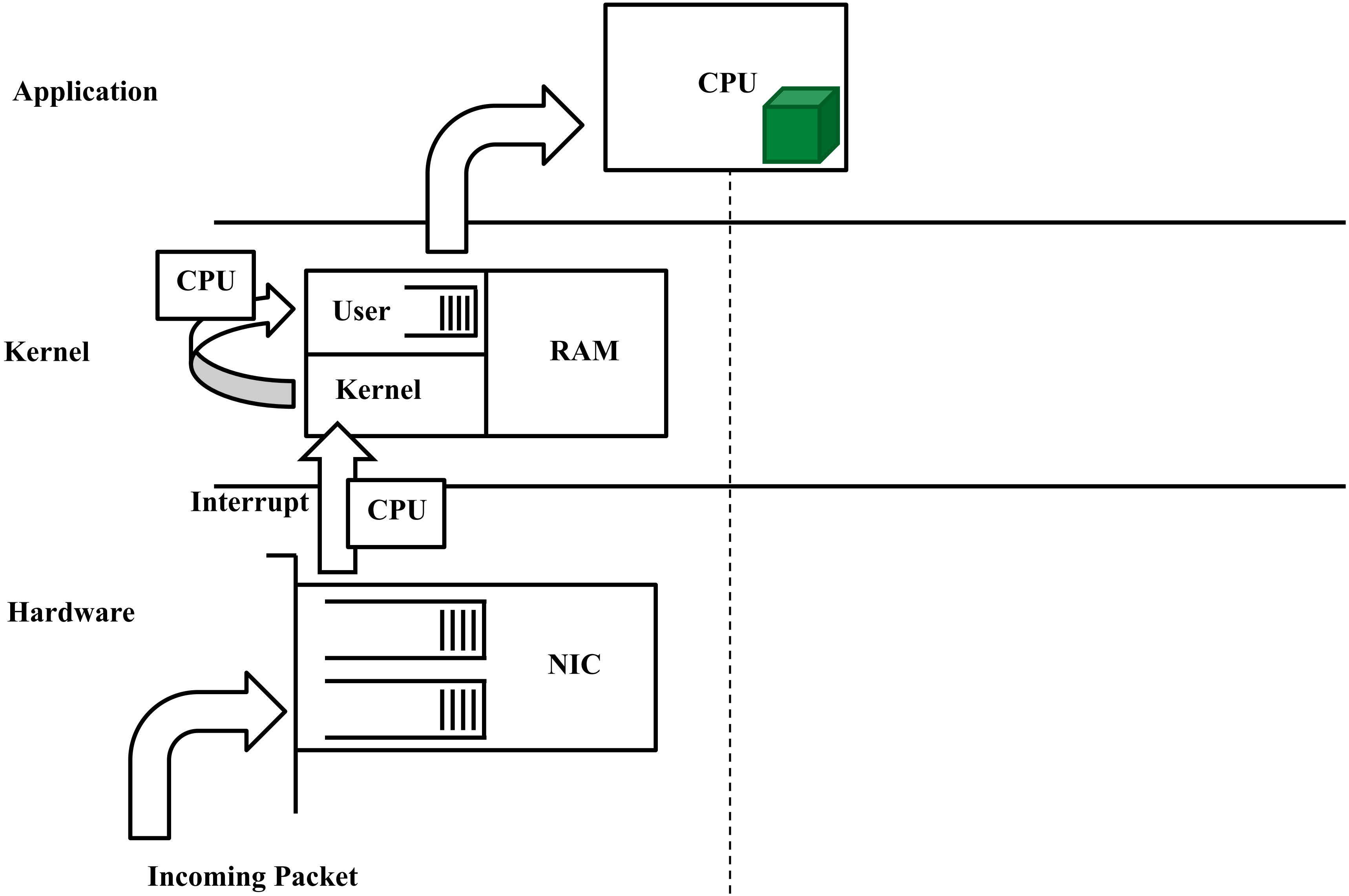
“Life of a Packet”



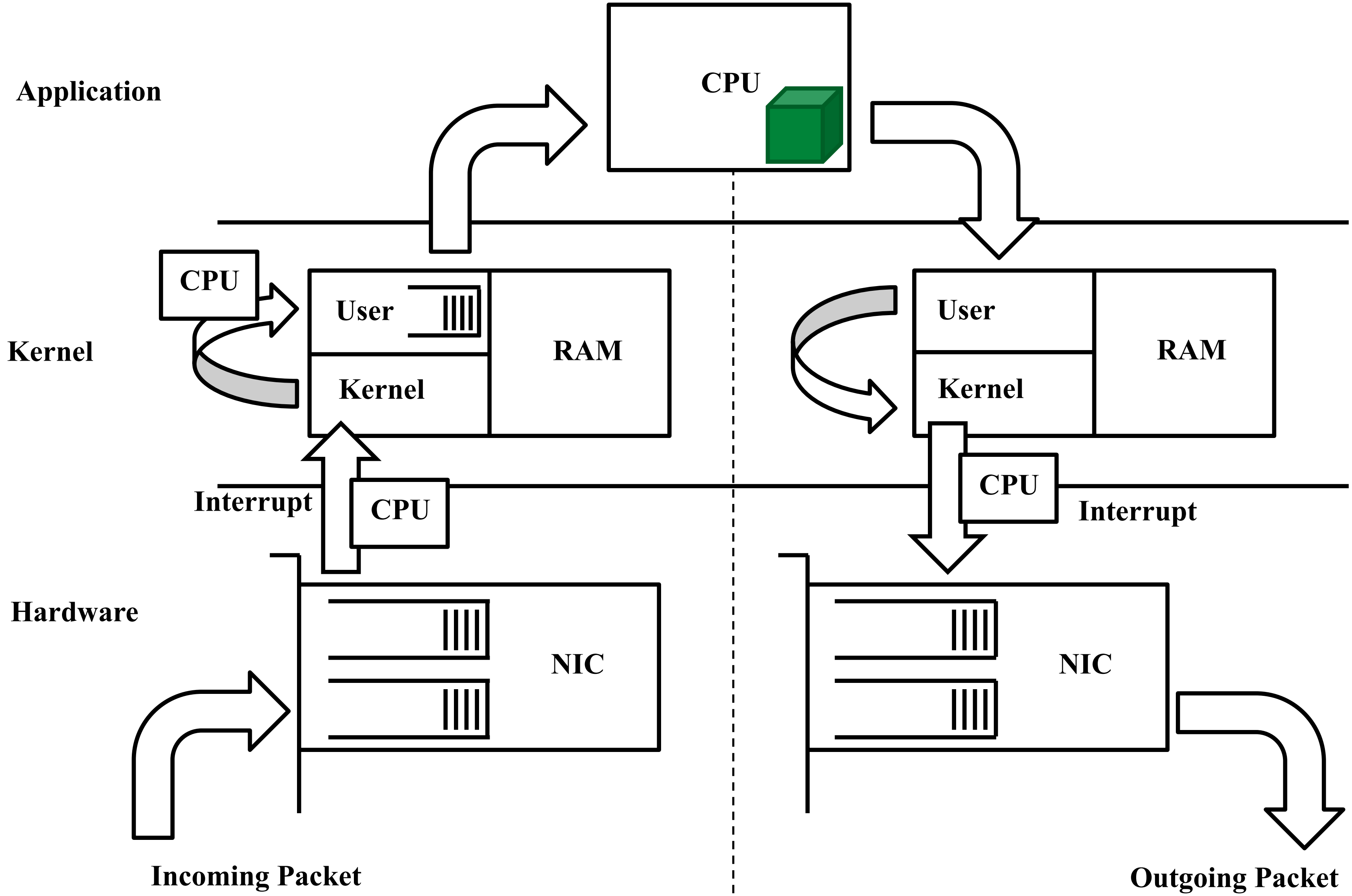
"Life of a Packet"



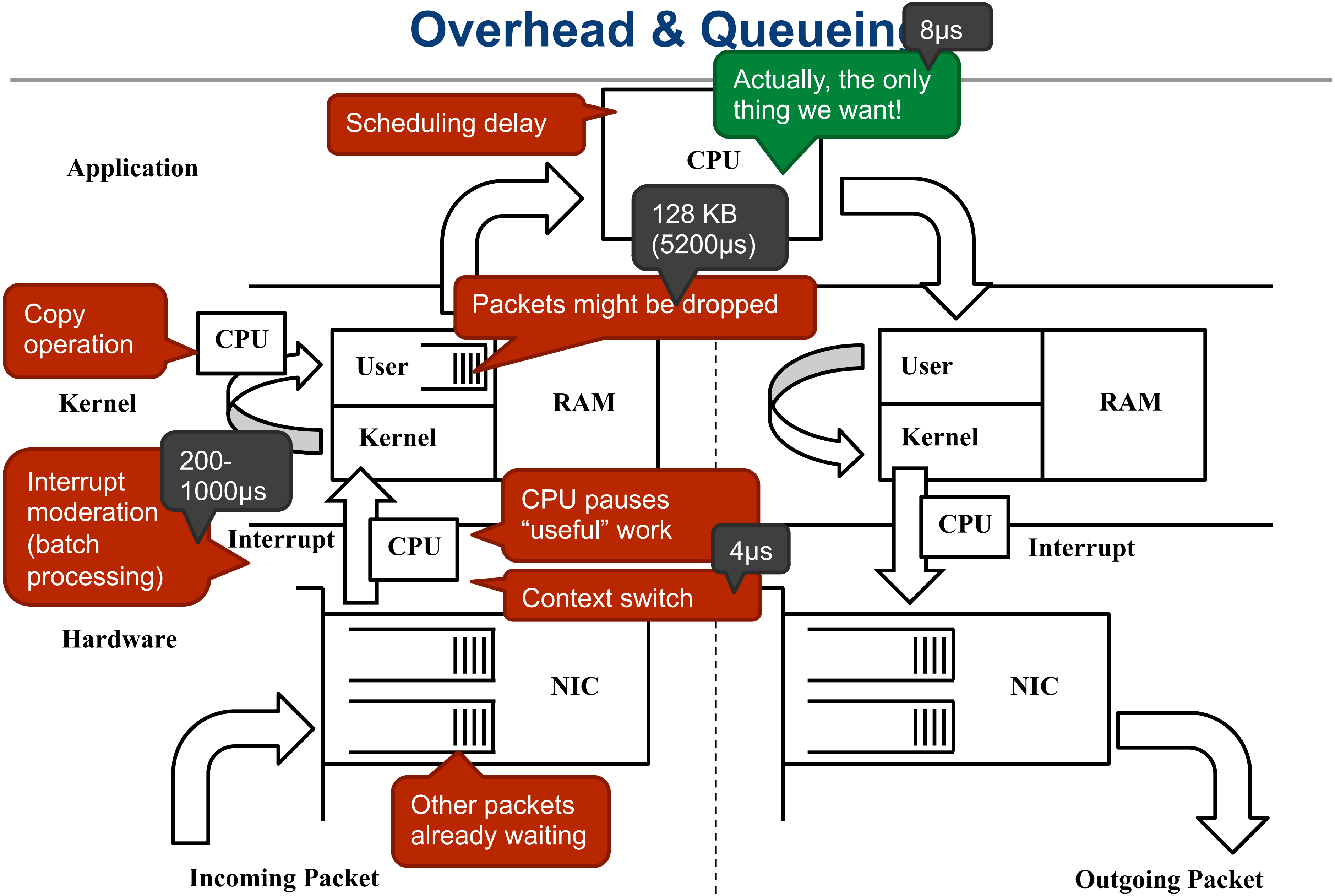
“Life of a Packet”



“Life of a Packet”



Overhead & Queueing



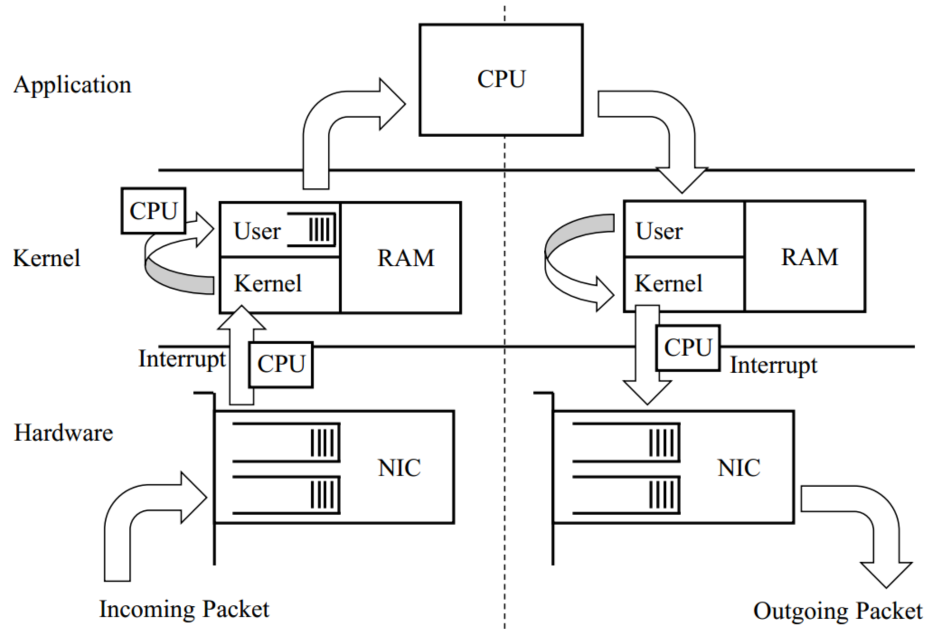
Steffen Gebert, *Thomas Zinner*

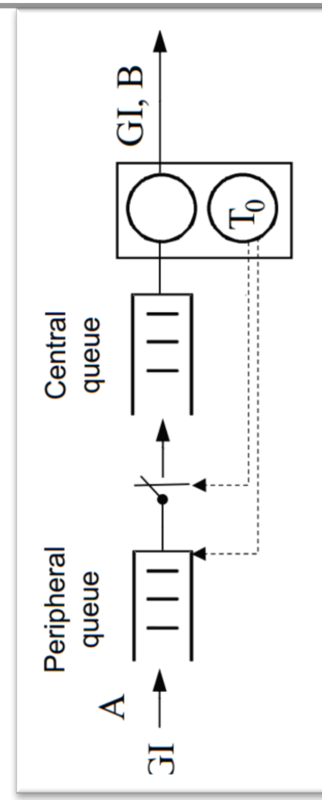
Can we derive an

ANALYTICAL MODEL?

Packet Processing in a Server

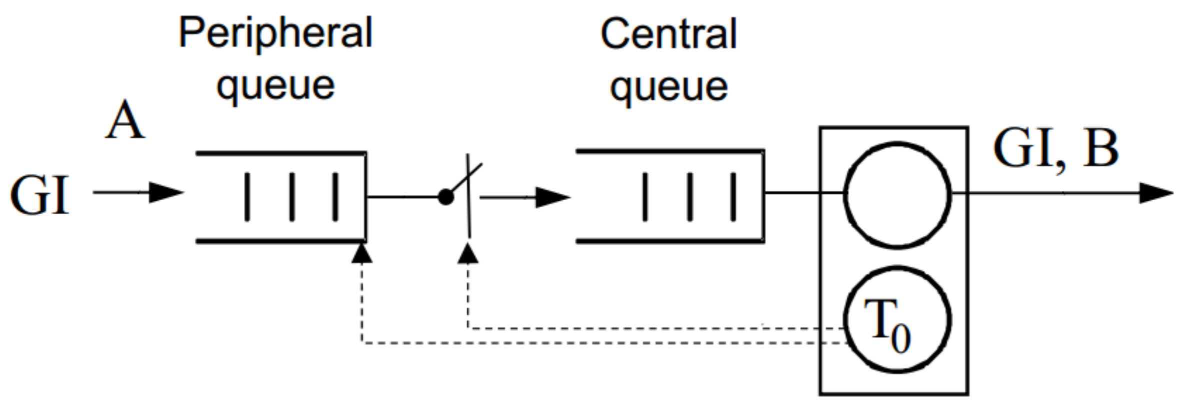
- Abstractions welcome!





- Interrupt moderation: Packet aggregation at NIC queue
- Interrupt triggers copy operation to kernel space
- Copy operation to user space memory
- Packets stored in buffer before application requests them from OS

Queueing Model



- Packet arrivals follow an arbitrary distribution A
- Peripheral queue (NIC) with infinite size
- Batch processing of packets
  - First packet after a batch transfer initiates timer τ (fixed interval length[1])
- Central queue (CPU / SW) modeled as a $GI^{[X]}/GI/1-S$ system

1) `ethtool -C rx-usecs 1000`

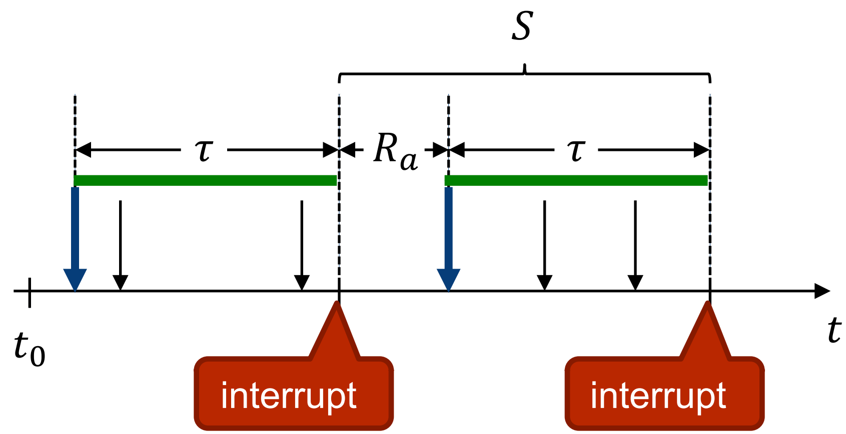
Peripheral Queue (NIC) – Model

- Batch interarrival time s
  - $s(k) = r_a(k) * \tau(k) * o(-k)$

recurrence time of packet arrival

aggregation interval

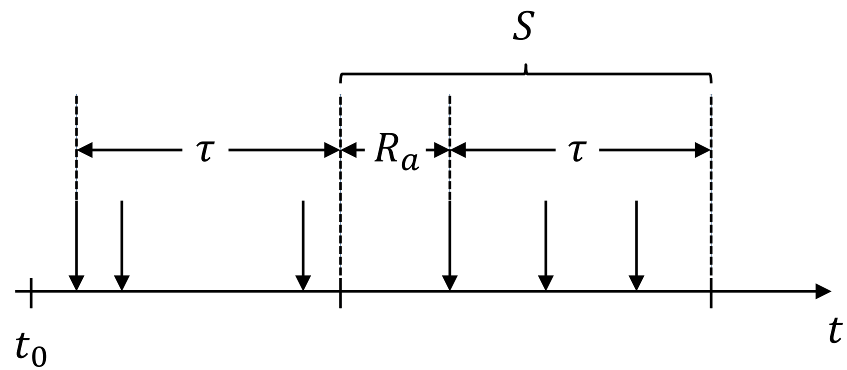
interrupt overhead



Peripheral Queue (NIC) – Model

- Batch interarrival time s
  - $s(k) = r_a(k) * \tau(k) * o(-k)$

recurrence time of packet arrival | aggregation interval | interrupt overhead

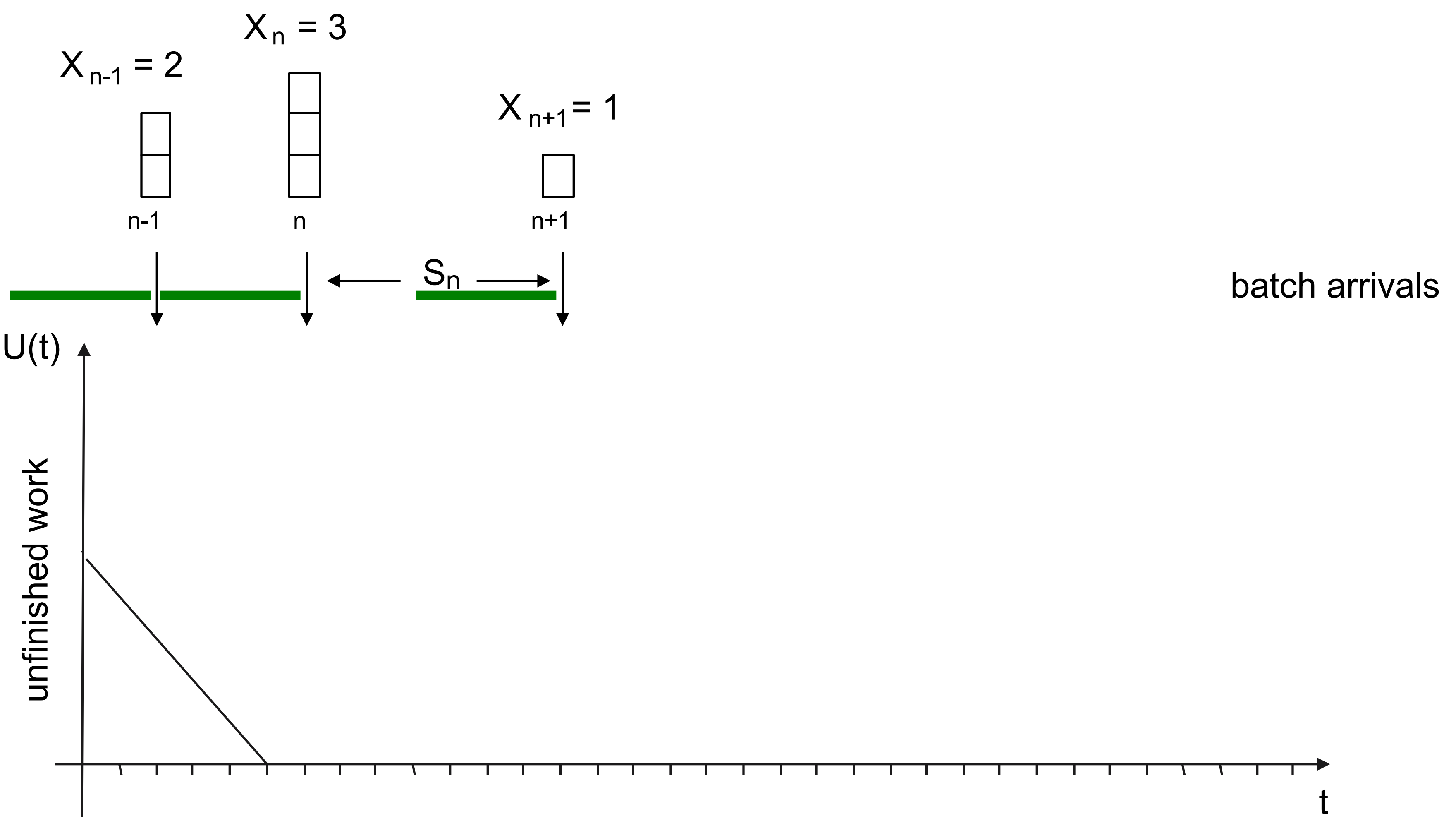


- Waiting time distribution of i-th packet w_i

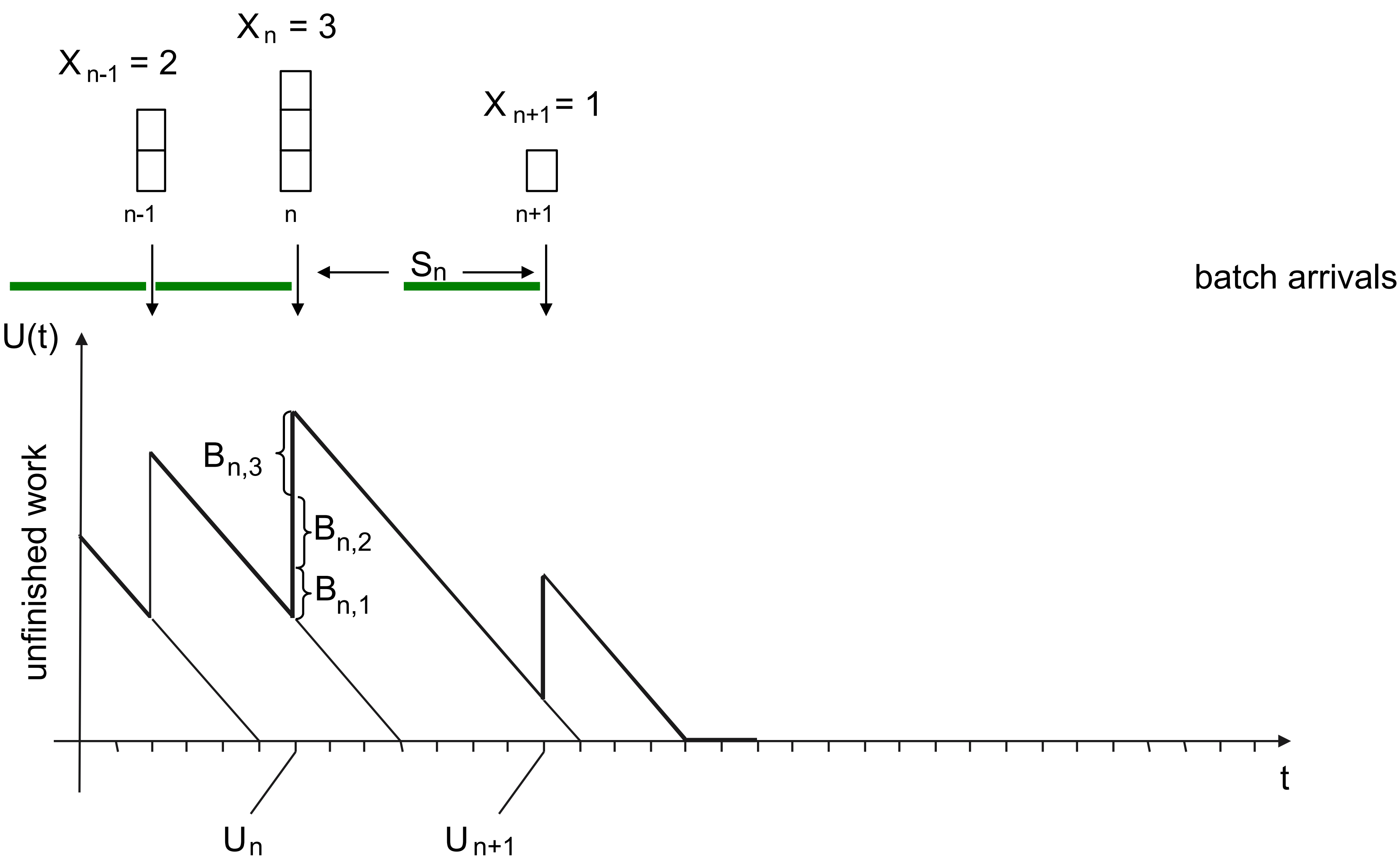
sweep operator | aggregation interval | packet inter-arrival

  - $w_i(k) = \pi_0[\tau(k) * \underbrace{a(-k) * \cdots * a(-k)}_{(i-1) \text{ times}}]$
  - The first packet in a batch initiates the aggregation interval
  - The later a packet arrives during an interval, the shorter its waiting time

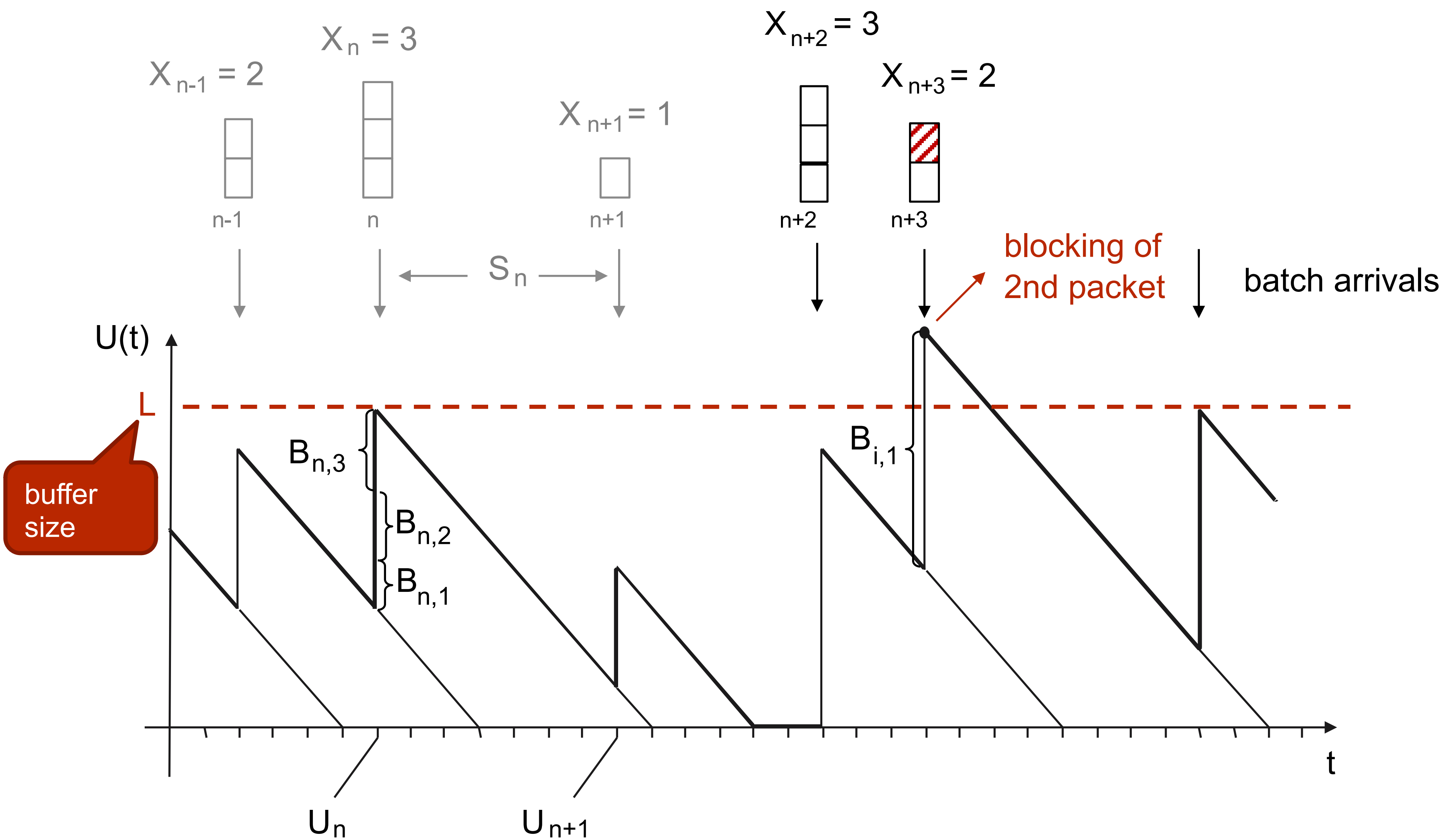
Central Queue – Exemplary Development



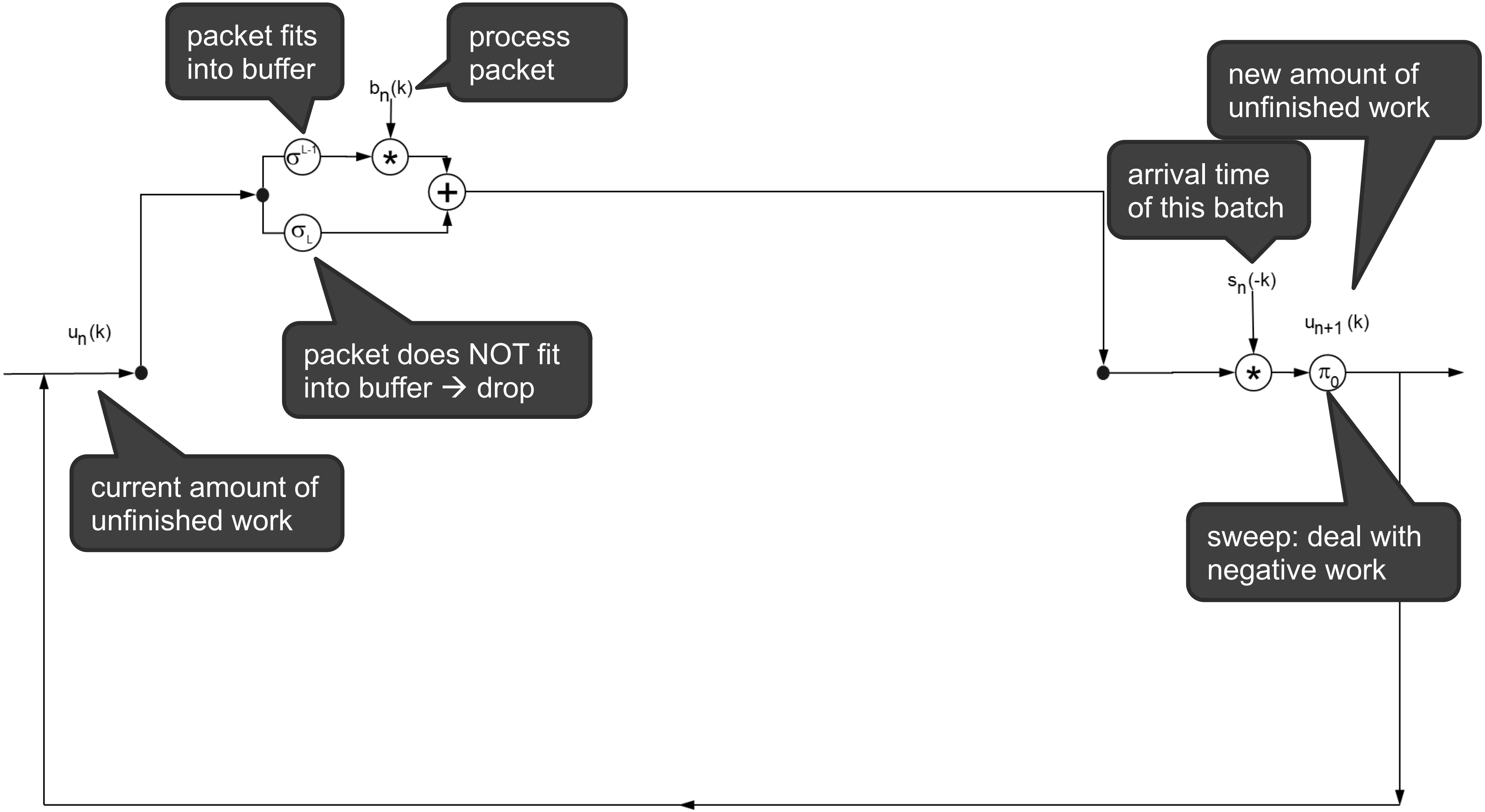
Central Queue – Exemplary Development



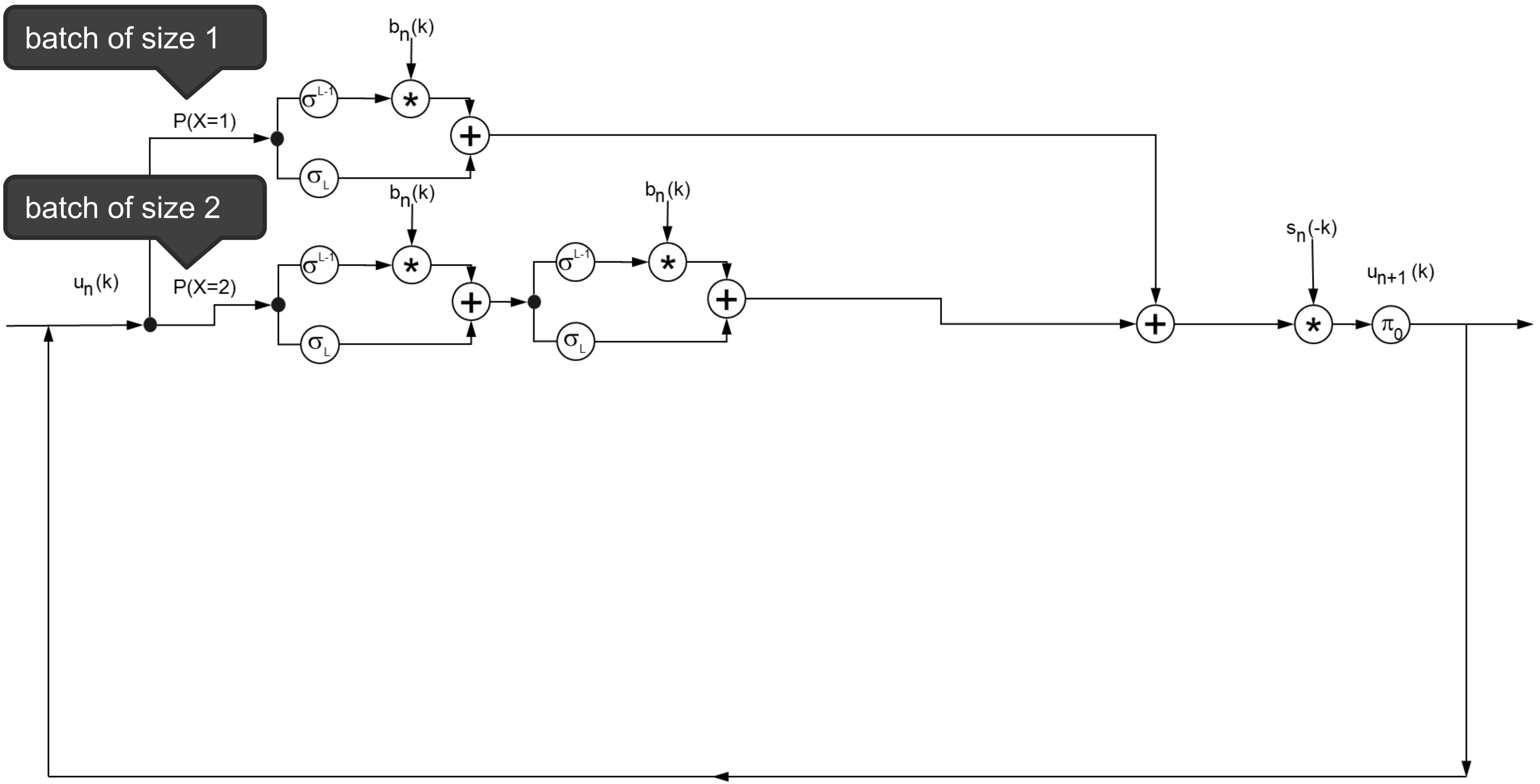
Central Queue – Exemplary Development



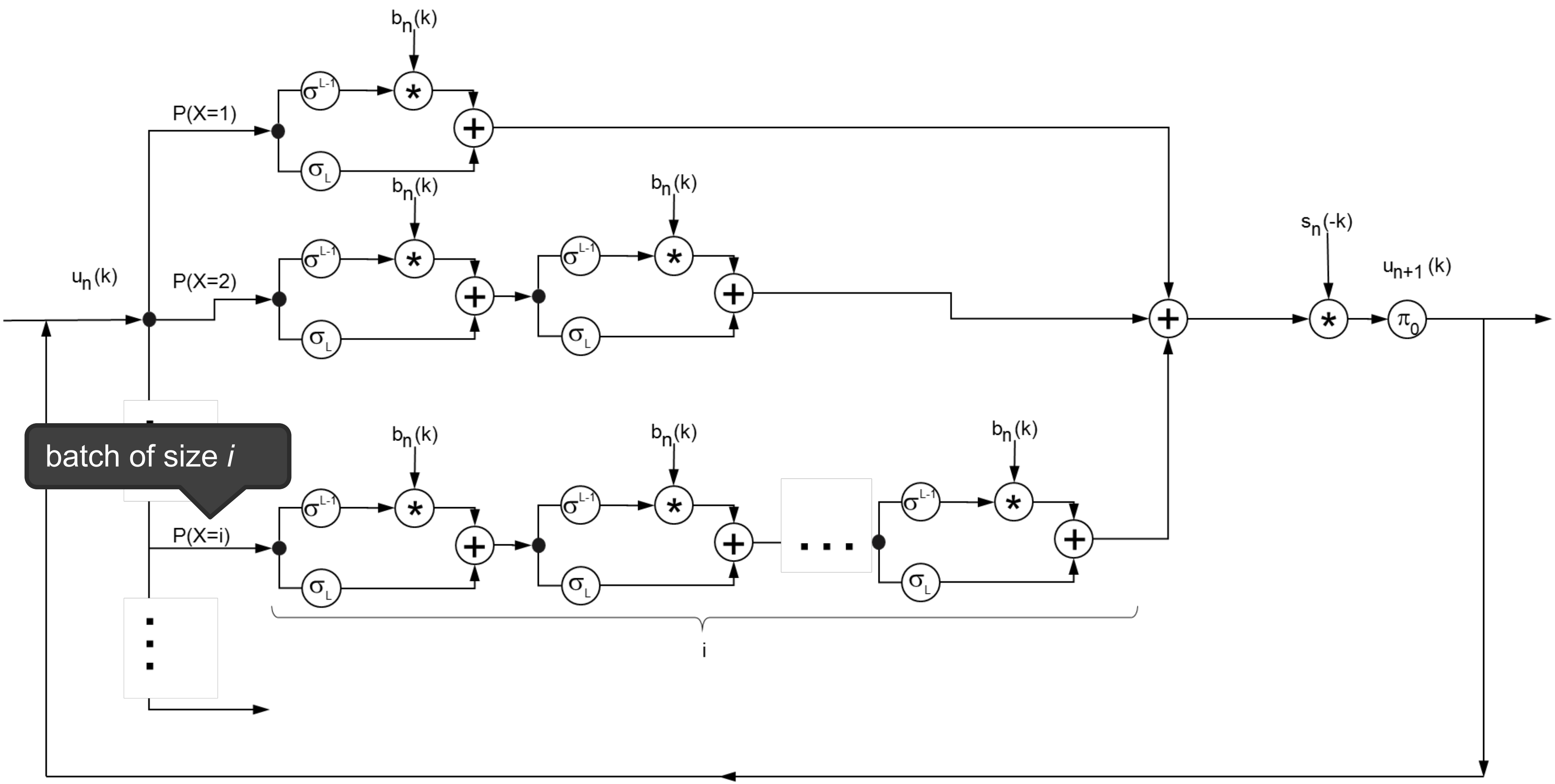
Central Queue – Computational Diagram



Central Queue – Computational Diagram



Central Queue – Computational Diagram



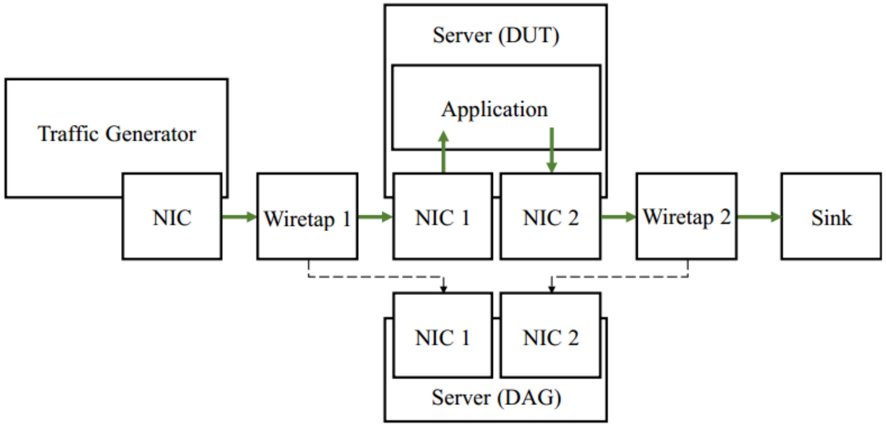
- In case of stationary system: $U = \lim_{n \to \infty} U_n$

Applicability of the Model

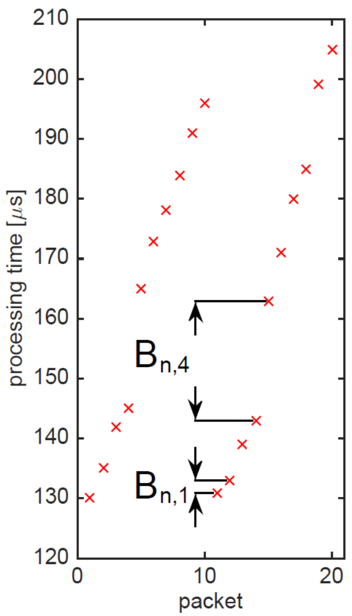
VALIDATION

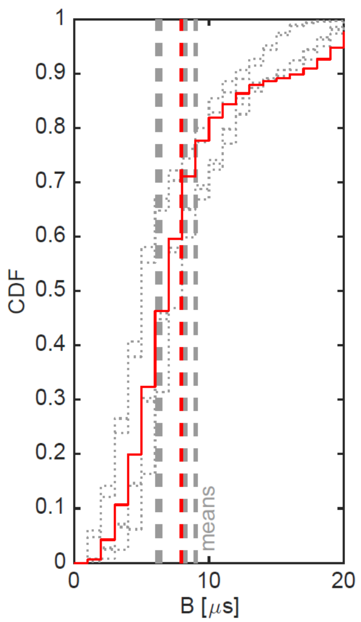
Testbed Setup and Service Time Distributions

- Local testbed to
  - Measure service times
  - Check applicability of derived model

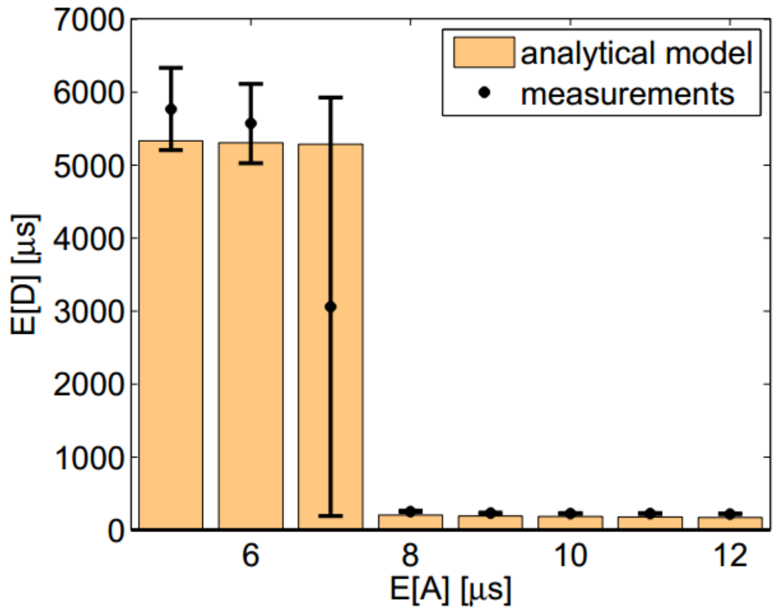


- Service time measurements
  - Generation of GTP-U packets every 100 µs
  - Aggregation interval length τ =1ms
  - Timestamps captures with *tcpdump*
  - Several repetitions to explore variability

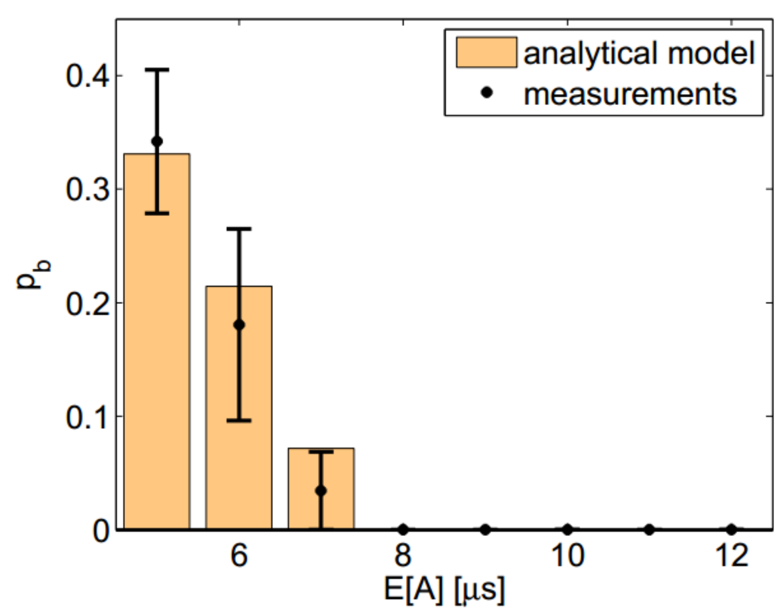




Applicability of the Model



Average processing times.



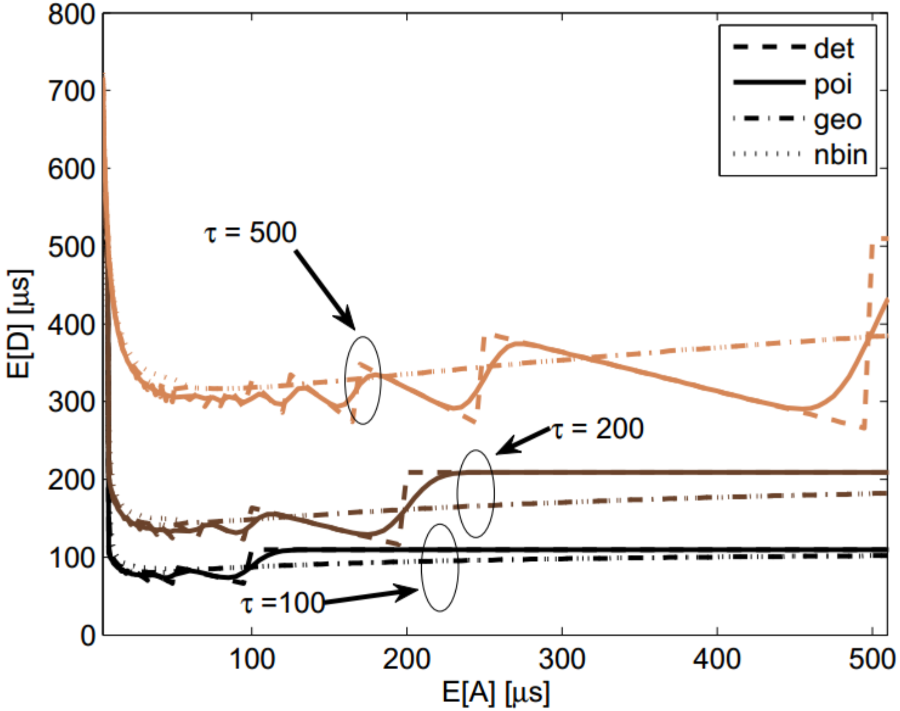
Packet loss.

- Processing times
  - Overlap between measurements and model for $E[A] < 8\mu s$
  - Only slight difference for $E[A] \geq 8\mu s$
- Packet loss
  - Overlap in all cases except for $E[A] = 7\mu s$ → Transition to overload

Influence of

SYSTEM PARAMETERS

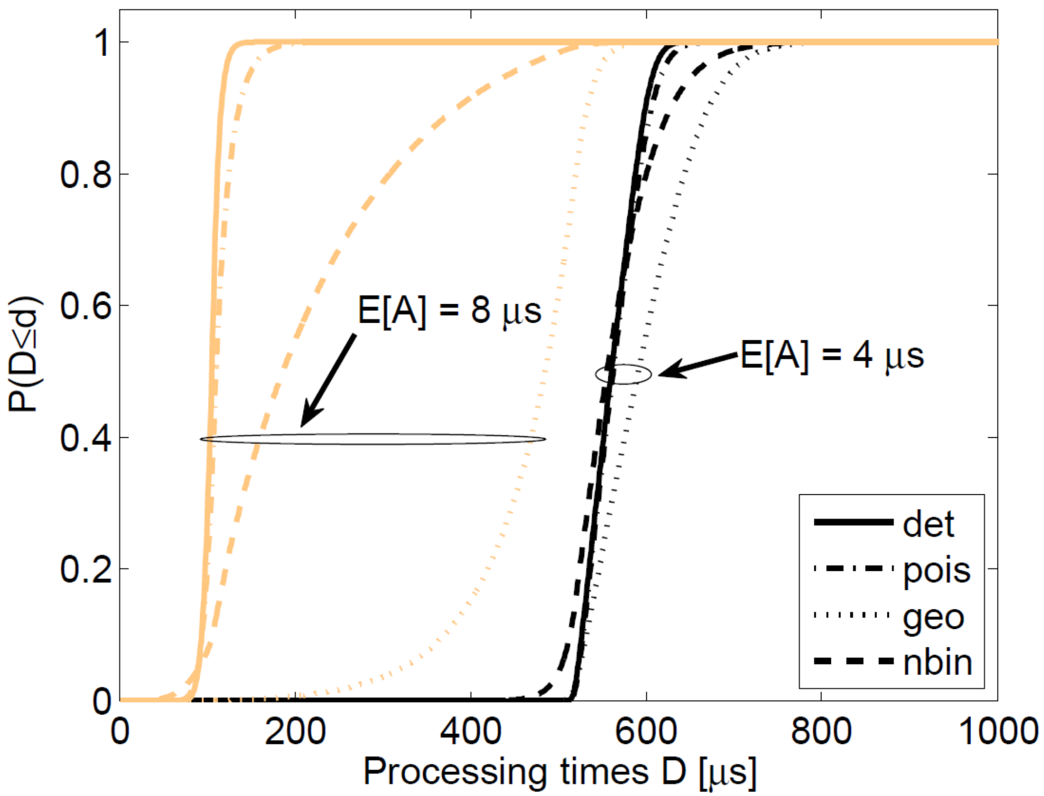
Influence of System Parameters on Performance



- Three phases
  - Small packet interarrival time → High processing time (waiting in CQ)
  - $E[A] > \tau$ → Most batches consist of one packet → Processing time $> \tau$
  - In-between: batches consist of several packets → Shorter waiting time in PQ → shorter total processing time

Processing Time Distributions

- Computation of processing time distributions for τ=100μs



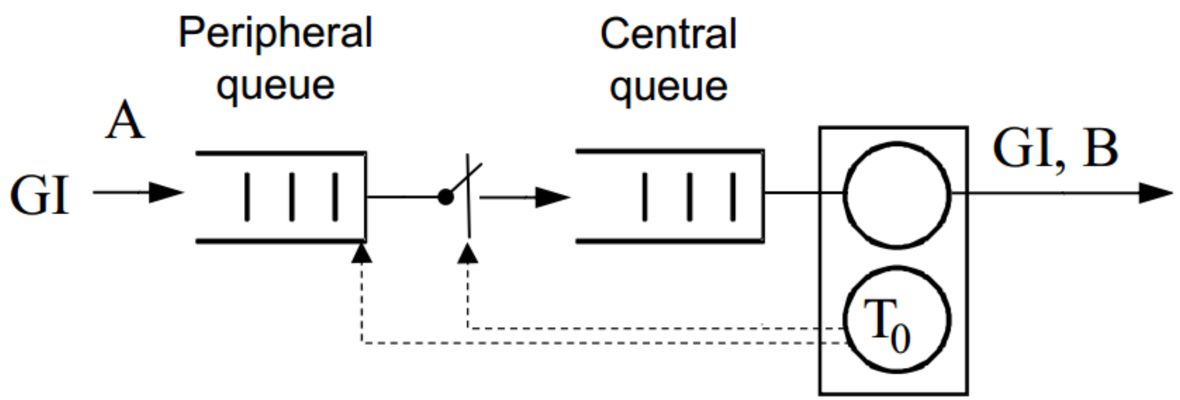
- Detailed investigation of the impact of different input parameters on the processing times possible

Summary & Conclusion

- Analytical model for analyzing performance implications of interrupt mitigation settings
- Determine different performance indicators
  - Service time including distribution
  - Packet loss probability
- Determine effects of different parameters on performance indicators
  - Interarrival times and their distributions
  - Length of aggregation intervals
- Benefits
  - Proper dimensioning of NFV-based systems
  - Performance prediction
- Comparison with testbed-based measurements indicates applicability

BACKUP

Queueing Model



- Packet arrivals follow an arbitrary distribution A
- Peripheral queue (NIC) with infinite size
- Batch processing of packets
  - First packet after a batch transfer initiates timer τ
- Central queue (CPU / SW) modeled as a $GI^{[X]}/GI/1-S$ system

Sum of Discrete Random Variables

- X as the sum of two non-negative discrete random variables X_1 and X_2

$$X = X_1 + X_2$$ (X_1 and X_2 statistically independent)

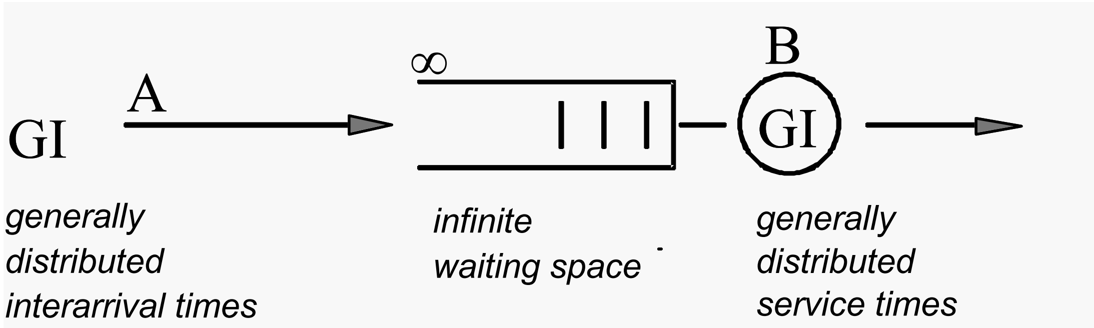
- Distribution of X:

$$x(i) = P(X=i) = P(X_1 + X_2 = i)$$
$$= \sum_{j=0}^{i} P(X_1 = i-j \mid X_2 = j) \cdot P(X_2 = j)$$
$$= \sum_{j=0}^{i} x_1(i-j) \cdot x_2(j)$$

- Or: $$x(i) = x_1(i) * x_2(i)$$ (discrete convolution)

- Example: sum of two six-sided dice

Discrete-time Queue GI/GI/1



- Discrete-time Δt
- Interarrival time A and service time B are discrete-time random variables:

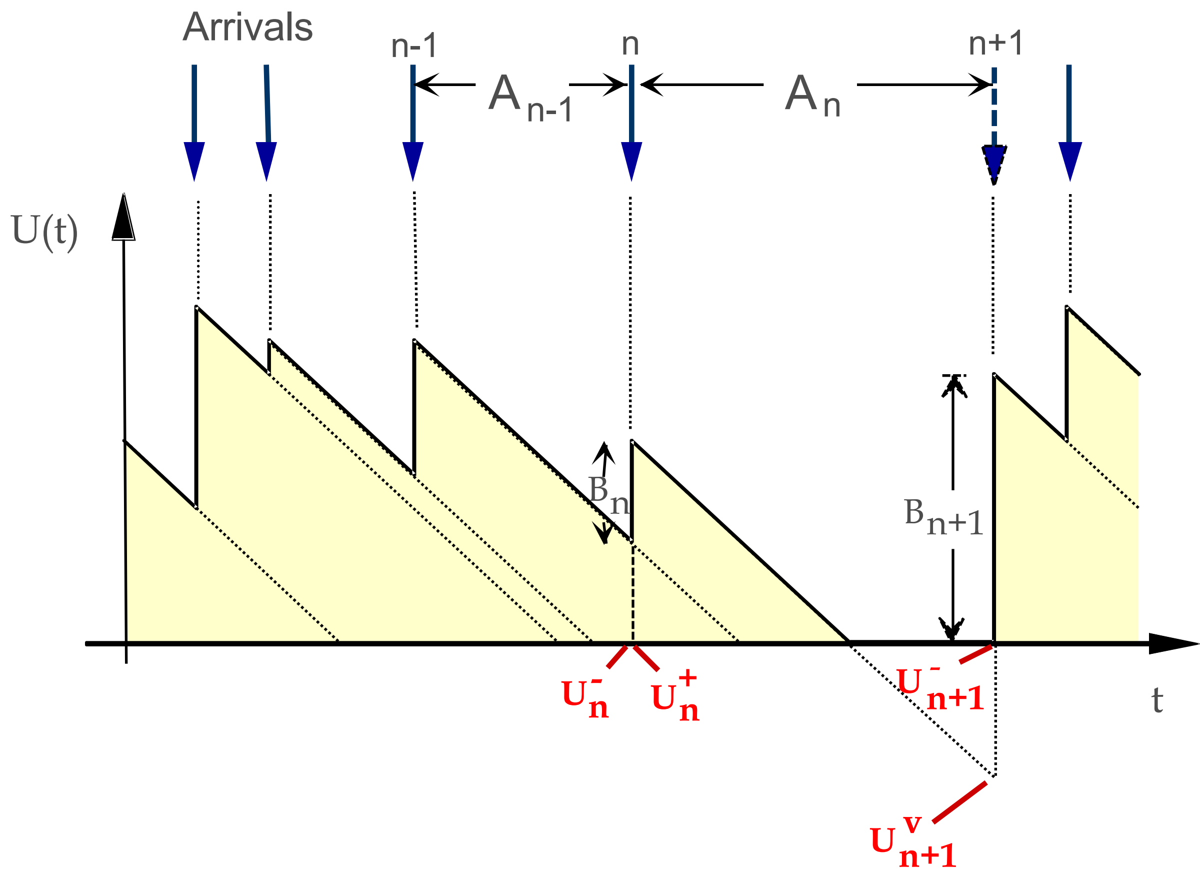
a(k) , k=0, 1, ...
Interarrival time distribution

b(k) , k=0, 1, ...
Service time distribution

- Utilization of the processing entity:

$$\rho = \frac{E[B]}{E[A]} =$$

System States



Time Domain Analysis

- Derivation of the single steps

$$U_n^- \xrightarrow[1]{} U_n^+ \xrightarrow[2]{} U_{n+1}^v \xrightarrow[3]{} U_{n+1}^-$$

- Step ① $U_n^+ = U_n^- + B_n \quad \rightarrow \quad u_n^+(k) = u_n^-(k) * b_n(k)$

- Step ② $U_{n+1}^v = U_n^+ - A_n \quad \rightarrow \quad u_{n+1}^v(k) = u_n^+(k) * a_n(-k)$

- Step ③ $U_{n+1}^- = \max(0, U_{n+1}^v) \rightarrow \quad u_{n+1}^-(k) = \pi_0(u_{n+1}^v(k))$

with $$\pi_m(x(k)) = \begin{cases} 0 & k < m \\ \sum_{i=-\infty}^{m} x(i) & k = m \\ x(k) & k > m \end{cases}$$

Time-Discrete Analysis of GI/GI/1

- Summary: $u^-_{n+1}(k) = \pi_0\left(u^-_n(k) * \underbrace{a_n(-k) * b_n(k)}_{c_n(k)}\right)$

  ➔ $$u^-_{n+1}(k) = \pi_0\left(u^-_n(k) * c_n(k)\right)$$

- First-Come-First-Serve service discipline:

  $W_n = U^-_n$ waiting time of the n-th request

- Recursive relationship for computing the waiting time

  $$w_{n+1}(k) = \pi_0\left(w_n(k) * c_n(k)\right)$$

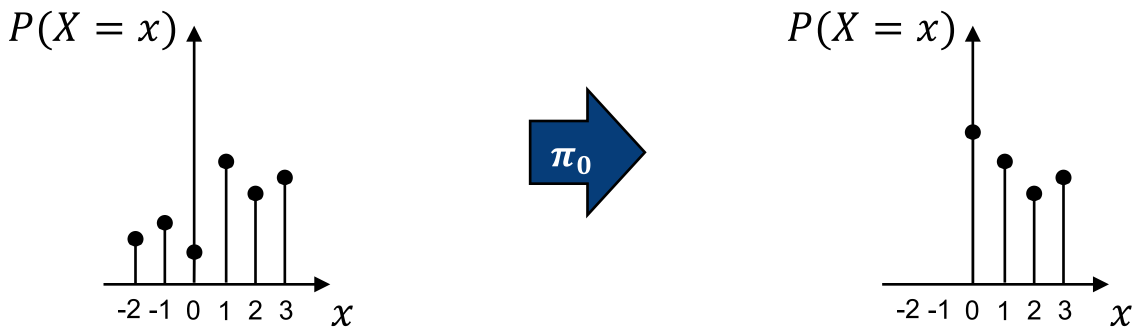
- In case of a stationary system

  $$A_n = A,\quad B_n = B,\quad W = \lim_{n\to\infty} W_n$$

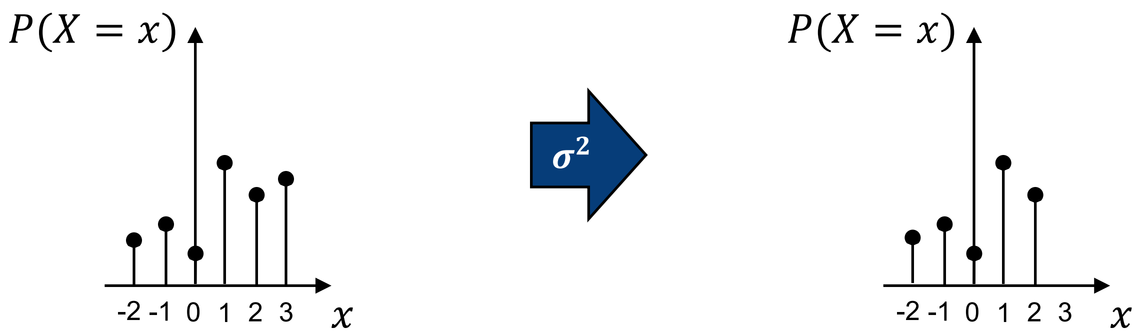
  ➔ $$w(k) = \pi_0\left(w(k) * c(k)\right)$$

Important Operators

- $\pi_0(\cdot)$
  - Sweep operator – sums the probability mass of negative unfinished work in the system and appends it to the state for an empty system.



- $\sigma^m(\cdot)$, $\sigma_m(\cdot)$
  - Operators that truncate the upper / lower part of a PDF



GI/GI/1: Computational Algorithm

- Based on

$$w_{n+1}(k) = \pi_0\left(w_n(k) * a_n(-k) * b_n(k)\right) = \pi_0\left(w_n(k) * c_n(k)\right)$$

the following algorithm for computing the waiting time distribution can be applied: