Self-sufficient Sensor Networks: from Energy Profiling to Optimal Sustainable Performance

Nicola Bui, nicola.bui@imdea.org
(a joint work with Michele Rossi, rossi@dei.unipd.it)

Developing the Science of Networks

Who am I

- Studied Electronics and Telecommunications at the University of Ferrara

Who am I

- Studied Electronics and Telecommunications at the University of Ferrara
- Joined the SIGNET group in 2006 and working on Wireless Sensor Networks

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

institute imdea networks

Who am I

- Studied Electronics and Telecommunications at the University of Ferrara
- Joined the SIGNET group in 2006 and working on Wireless Sensor Networks
- CEO of Patavina Technologies between 2009 and 2013

PATAVINA TECHNOLOGIES

Who am I

- Studied Electronics and Telecommunications at the University of Ferrara
- Joined the SIGNET group in 2006 and working on Wireless Sensor Networks
- CEO of Patavina Technologies between 2009 and 2013
- Now with IMDEA Networks Institute in Madrid

About this work

- Started during the SWAP project (2010-2014)
- Goal: design of self-sufficient networked sensor devices
- Partners: Patavina Technologies, Consorzio Ferrara Ricerche, Centre Tecnològic Telecomunica-cions Catalunya, WorldSensing

Outline

- Motivation
- Optimization Framework
- Energy Source Model
- Optimal Operational Point
- Self-Sufficient Energy Policies
- Results

Motivation

- The no-free-lunch theorem applies to Energy-Harvesting Wireless Sensor Networks

- Challenge:

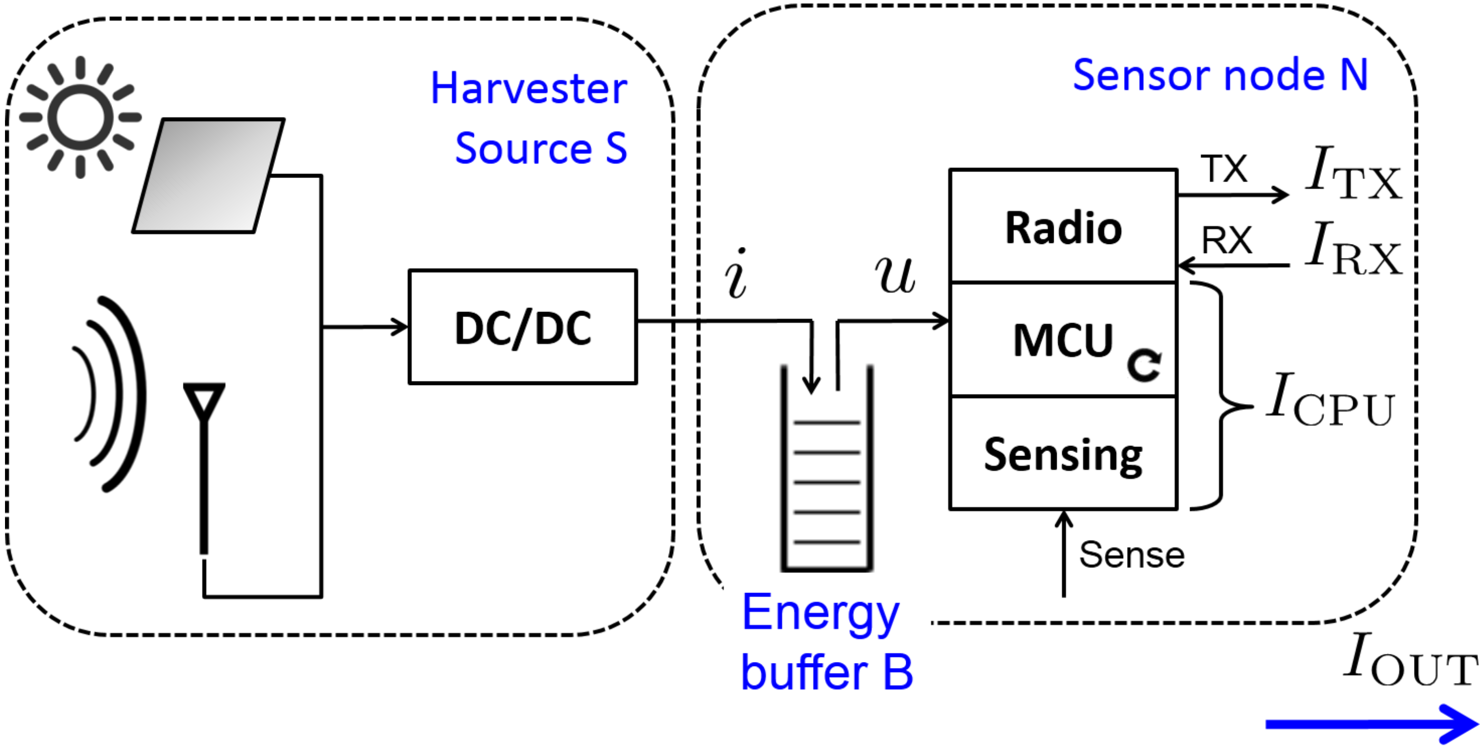
  Energy source is there but …

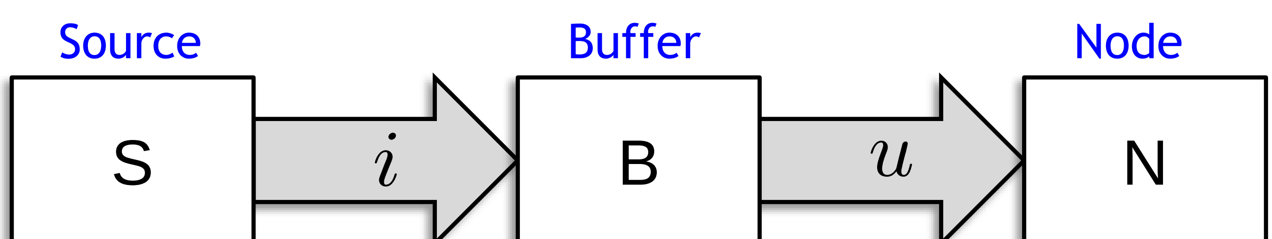
  … it is unreliable, erratic and intermittent

- Need for intelligent designs
  - Adaptive energy mangement
  - Transmission *vs* Storage *vs* Scavenger Size

Scenario (1) – Sensor Model

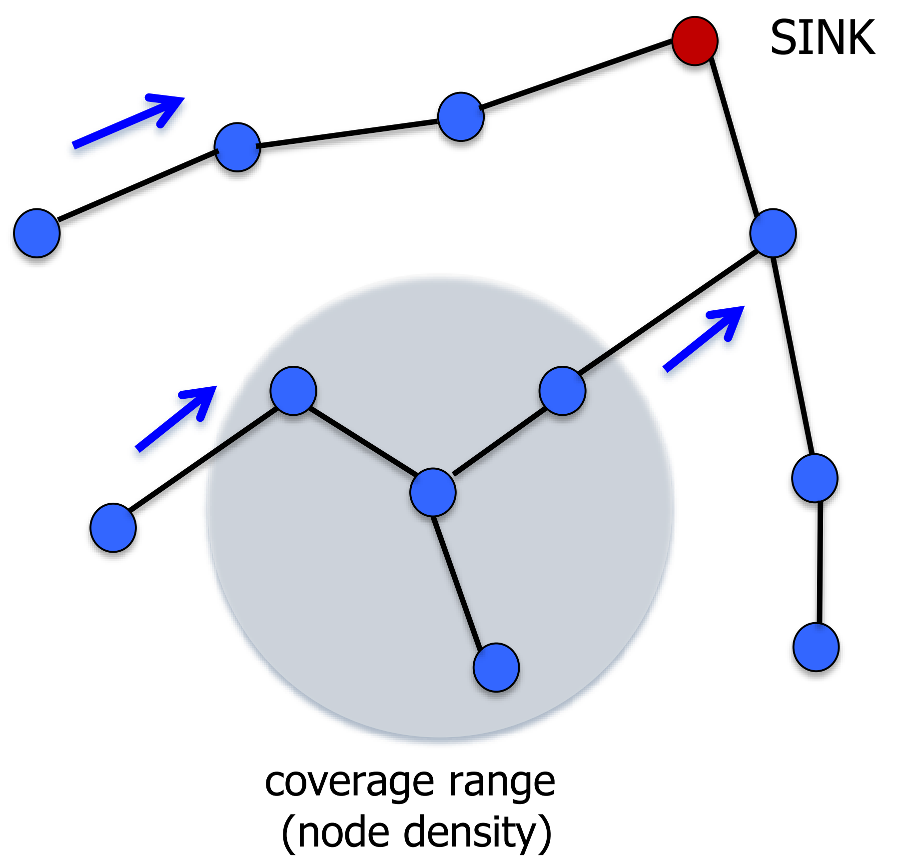
- Sensor: wireless device with sensing, processing and communication capabilities



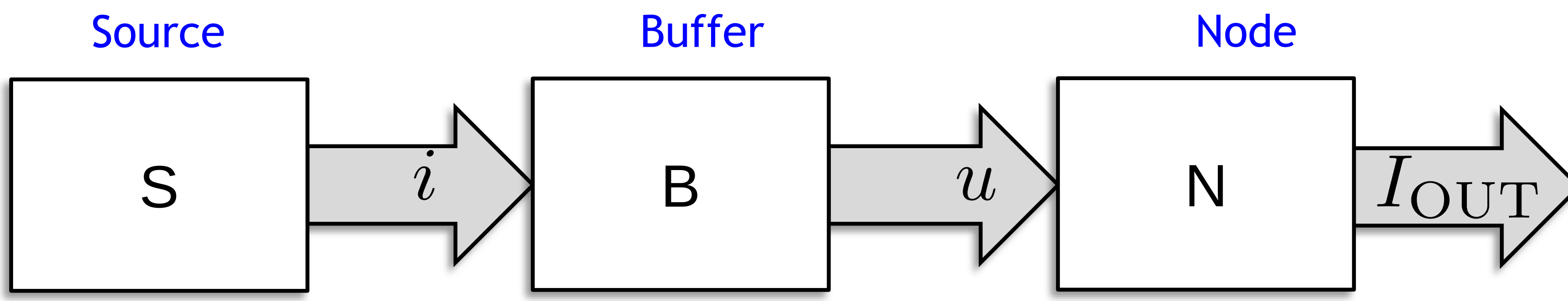


Scenario (2) – Network Model

- Scenario
  - Multi-hop routing
  - Data collection @ the sink
  - Energy harvesting nodes
- Aspects to model
  - MAC (channel access)
  - Routing
  - Energy consumption
  - Energy arrival



Optimization Framework



optimal working point for N

$$P1: \text{maximize}\, f_{\mathrm{U}}$$
$$\text{given } I_{\mathrm{OUT}} \le u$$

optimal energy-neutral policies

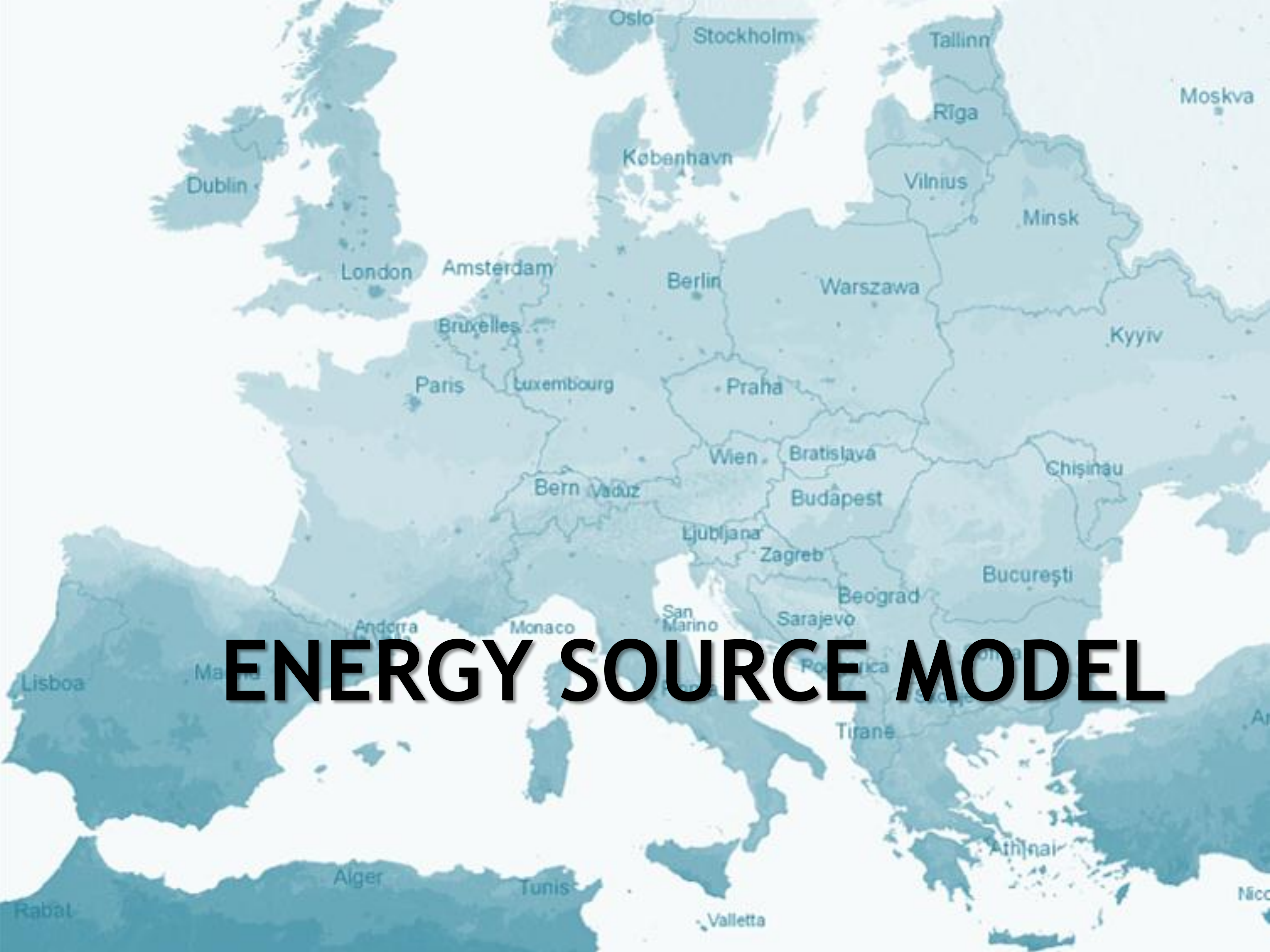
$$P2: \text{maximize reward}$$
$$\text{given source model } S$$

We optimize for the bottleneck node N → this assures network stability

Presentation Flow

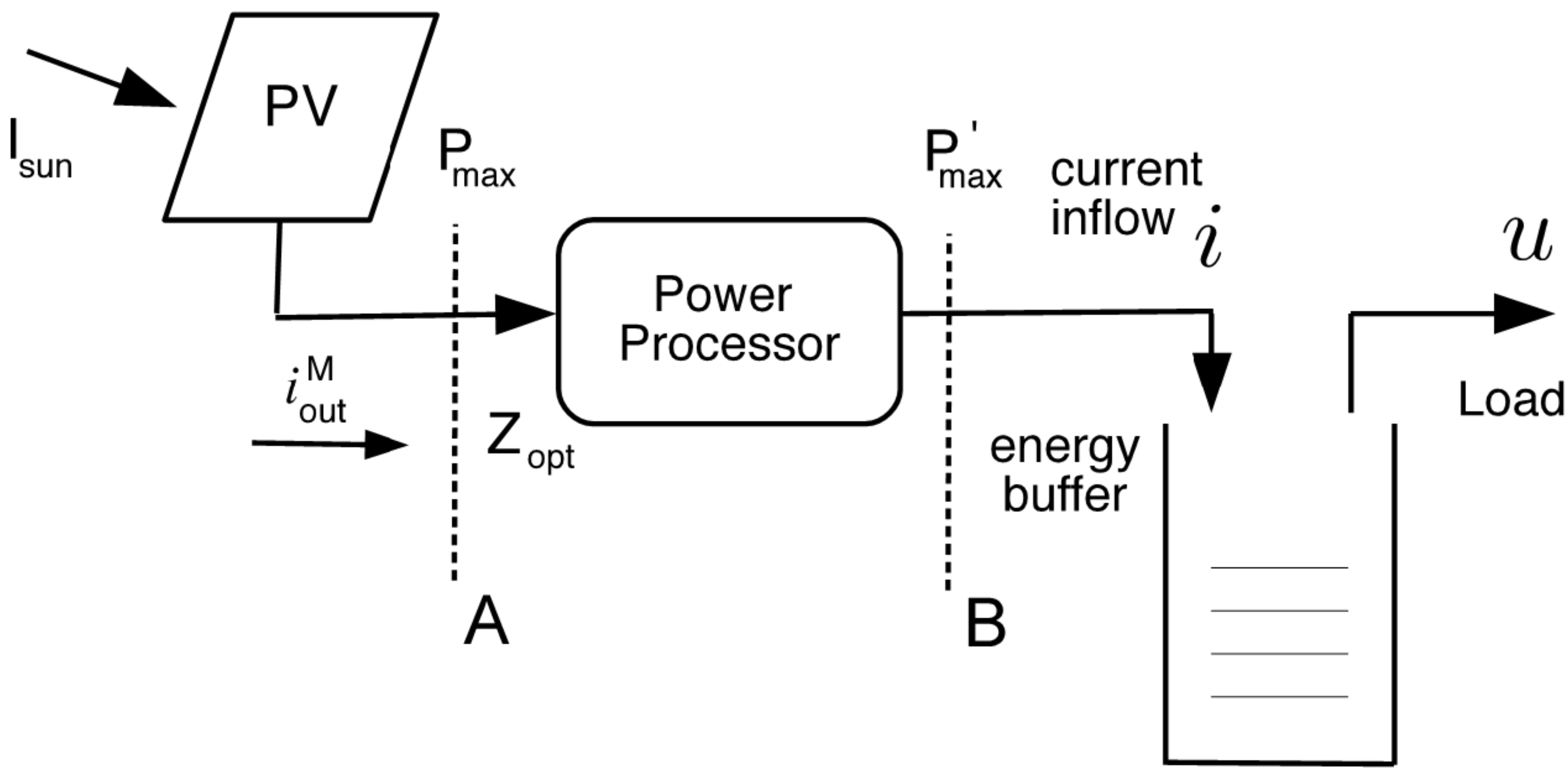
1) Energy Source Model, **S**
   - Input: harvester, energy availability
   - Output: current statistics $f_c(i|x_s)$
2) Optimal Operational Point, **P1**
   - Input: topology, MAC, hardware
   - Output: duty-cycle, throughput $(t^*_{\mathrm{U}}, t^*_{\mathrm{dc}})$
3) Energetically Self-Sufficient policies, **P2**
   - Input: energy source/consumption models
   - Output: energy management policies u

ENERGY SOURCE MODEL



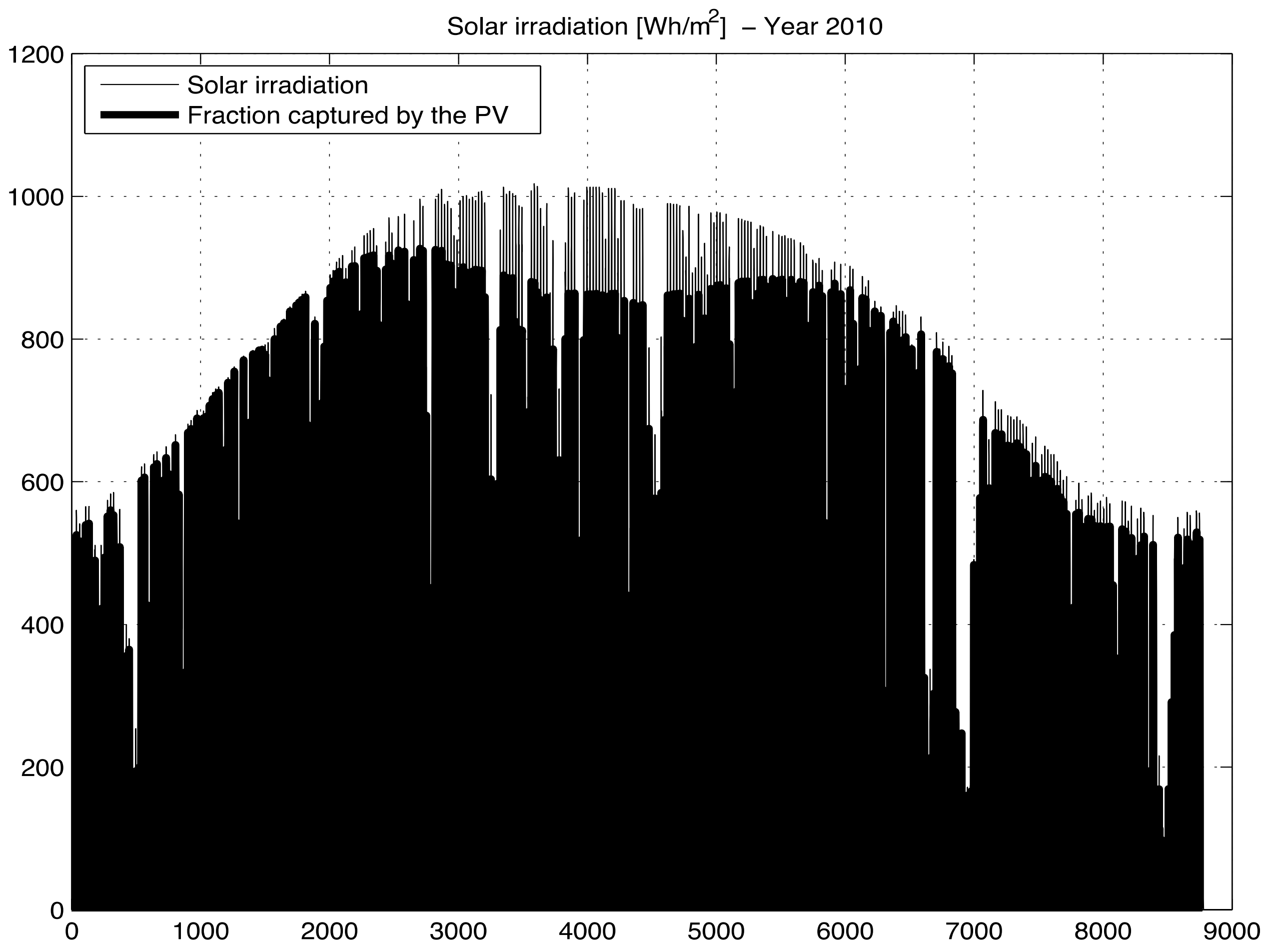
Energy Source Model

- Harvester: photovoltaic panel controlled by a processor to maximize the efficiency. The energy is stored in a battery.



Harvested Energy (1)

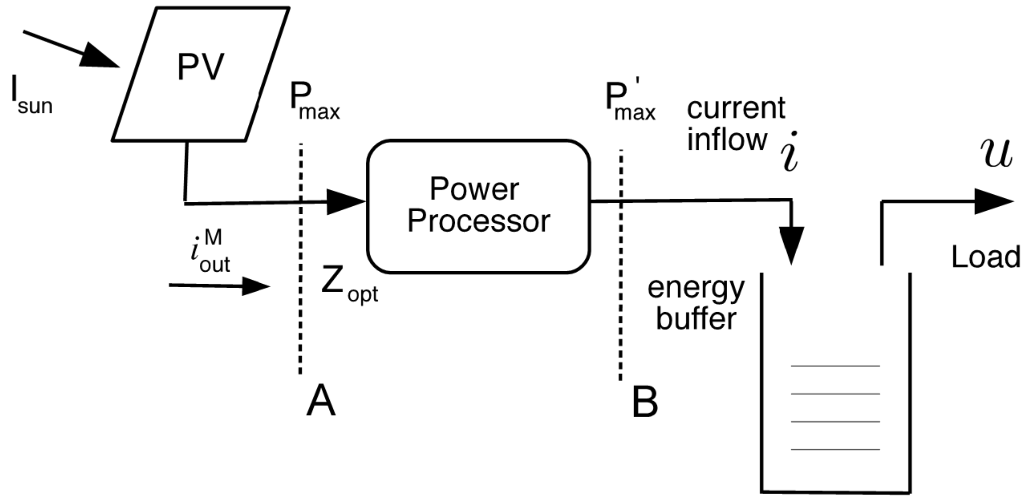
- Solar Irradiation for Los Angeles in 2010
  from NREL: http://www.nrel.gov/rredc/



NREL, National Renewable Energy Laboratory, "Renewable Resource Data Center," http://www.nrel.gov/rredc/

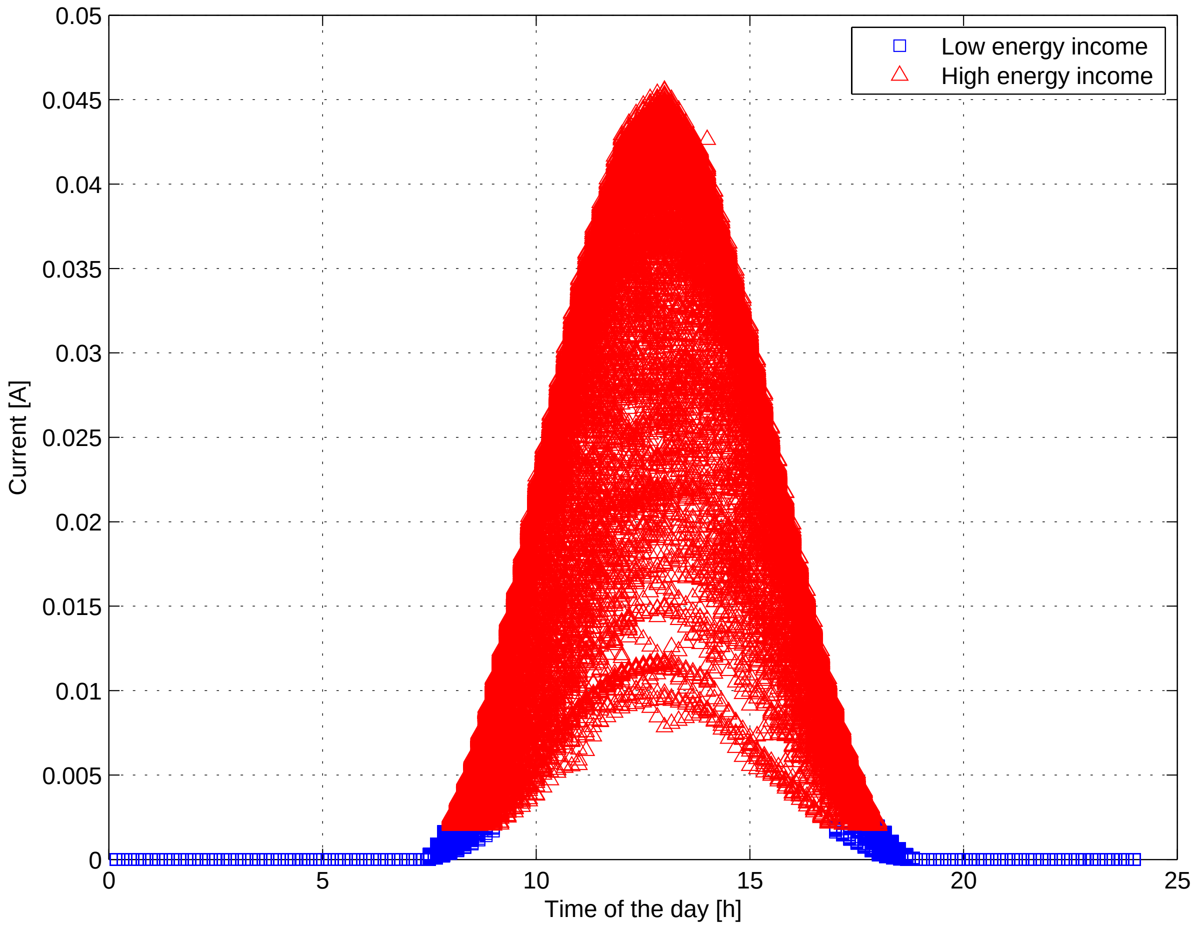
Harvested Energy (2)

- Solar radiation maps:
  - Latitude, longitude
  - Orientation of the panel
  - Day of year
  - Radiation maps
- PV technology:
  - Material
  - Efficiency
  - Panel size, tilt, azimuth
- DC/DC:
  - Efficiency
  - Optimal working point for the panel IV curve is assumed
- Statistical characterization of DC/DC out current
  - Current intensity [A]
  - Energy states (morning, afternoon, night, etc.)



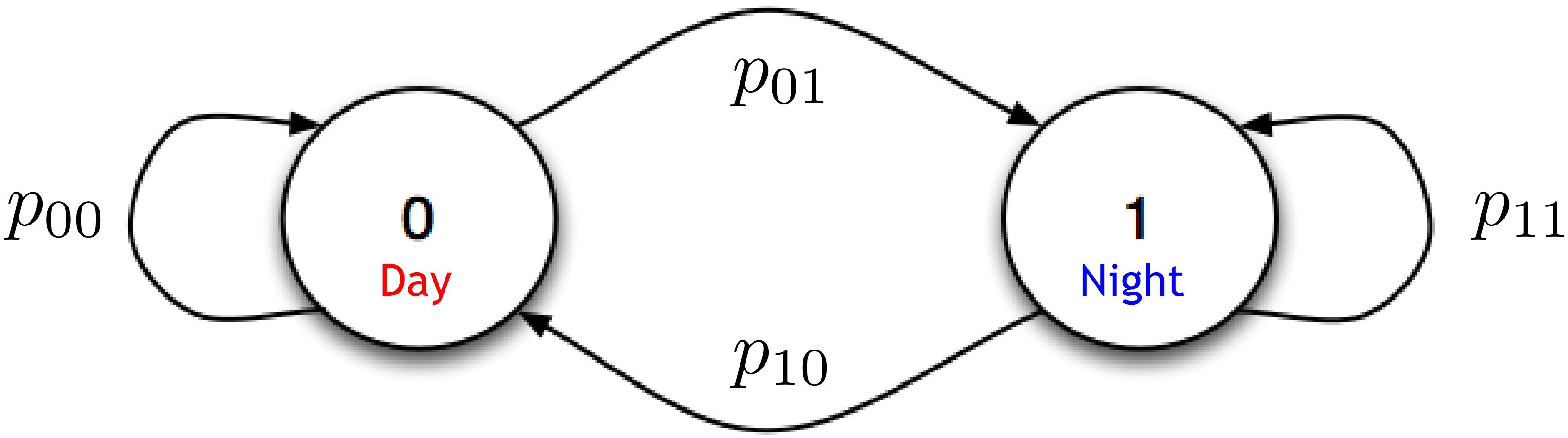
Example

- Statistics (pdf) of August (1999-2010)
  - Day/Night data clustering (duration and income)



Semi-Markov Model (SMM)

- Simplest 2-state model



Embedded chain probs

$$p_{10} = p_{01} = 1$$

$$p_{00} = p_{11} = 0$$

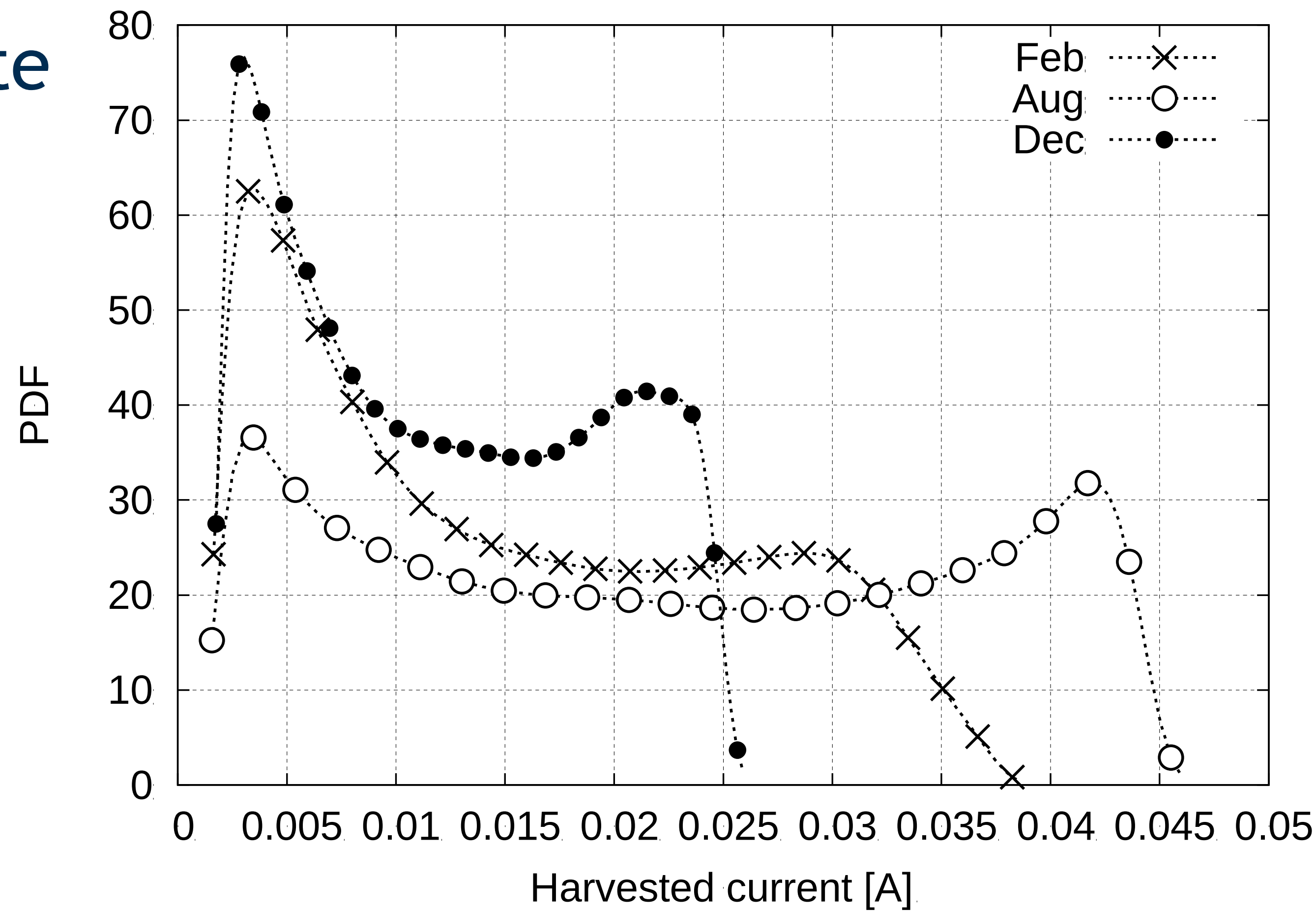
$f_c(i|x_s)$ Harvested current

$f_d(\tau|x_s)$ Permanence time

the approach has been generalized to any number of states

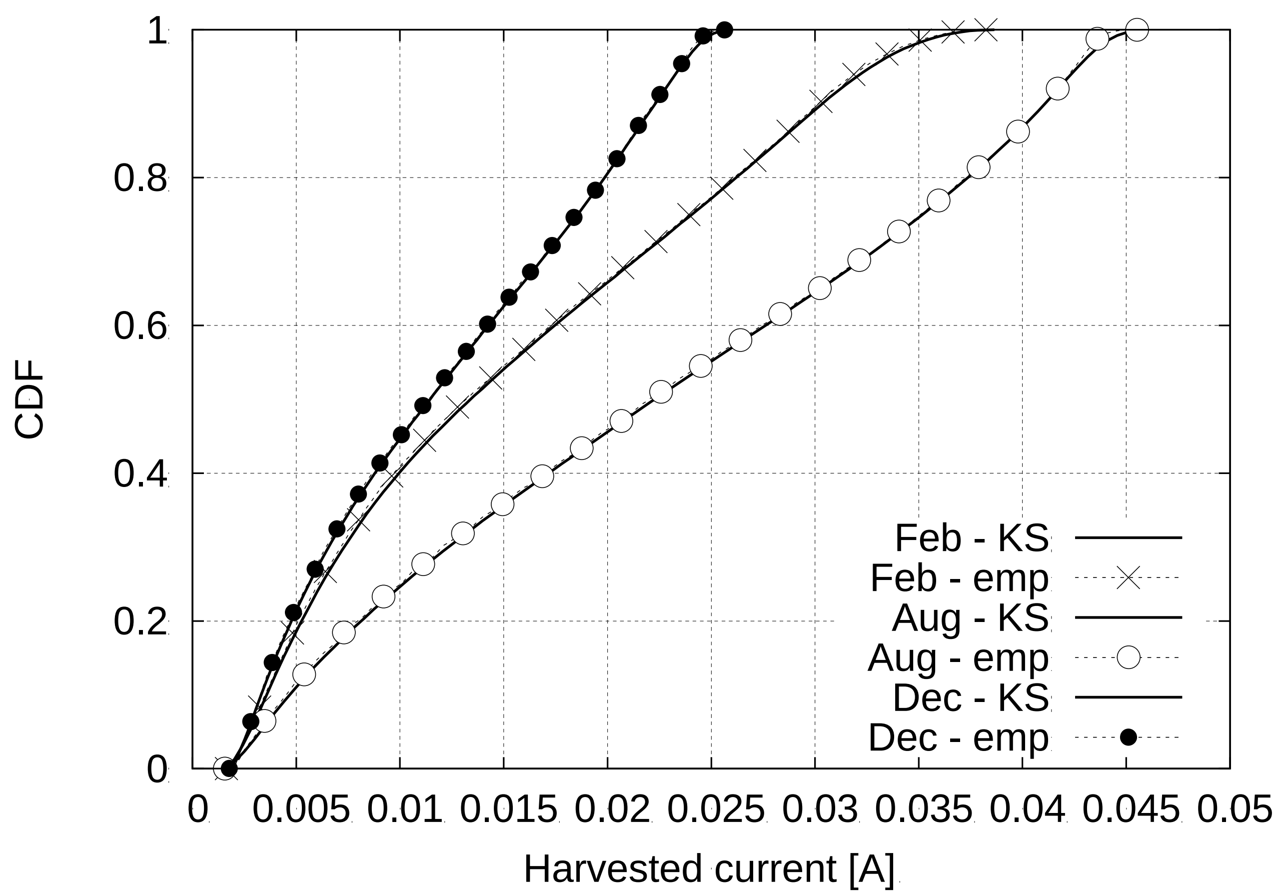
Statistics – Input Current $f_c(x|x_s)$

- Los Angeles data collected from 1999 to 2010
- Solar module: square 6x6 cm^2
- Day state



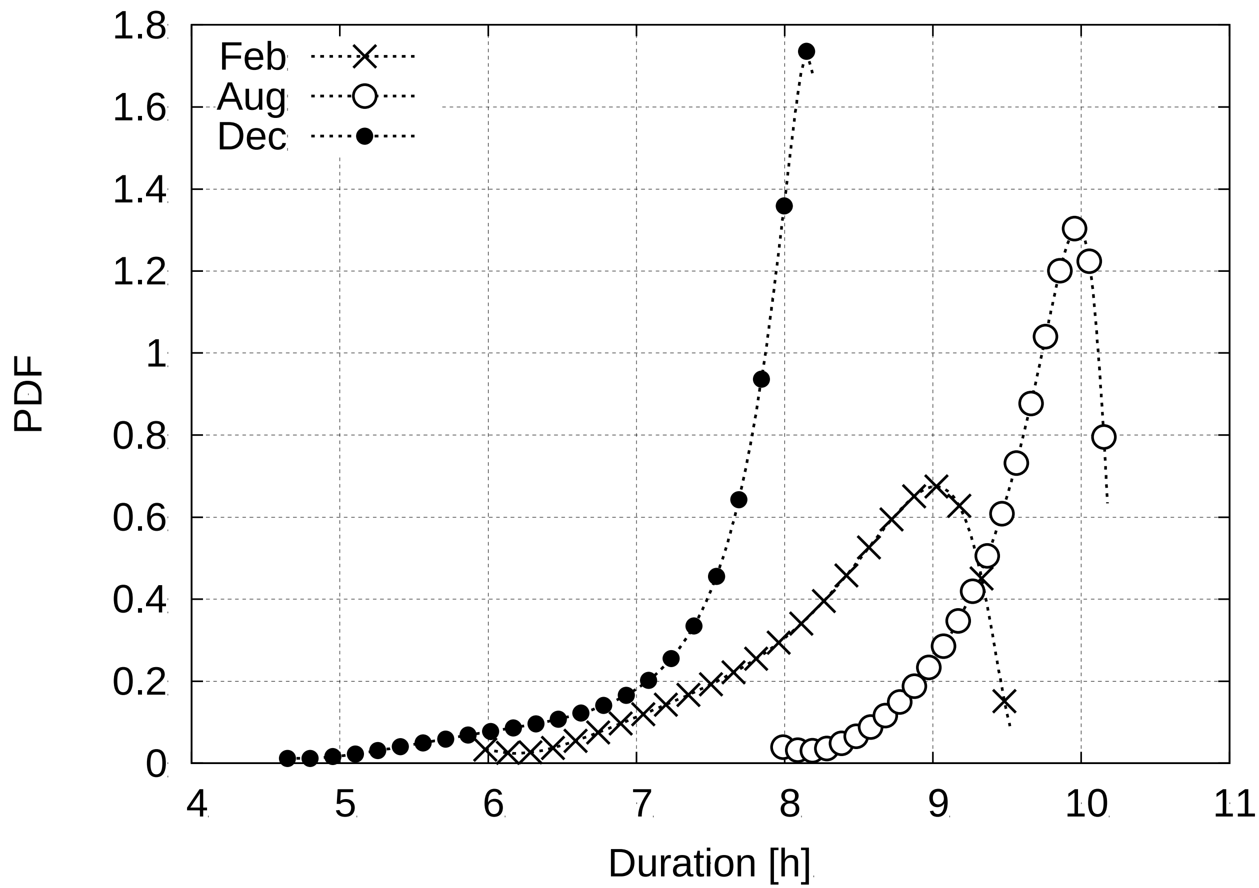
Statistics – Input Current CDF

- Los Angeles data collected from 1999 to 2010
- Solar module: square 6x6 cm^2
- Day state



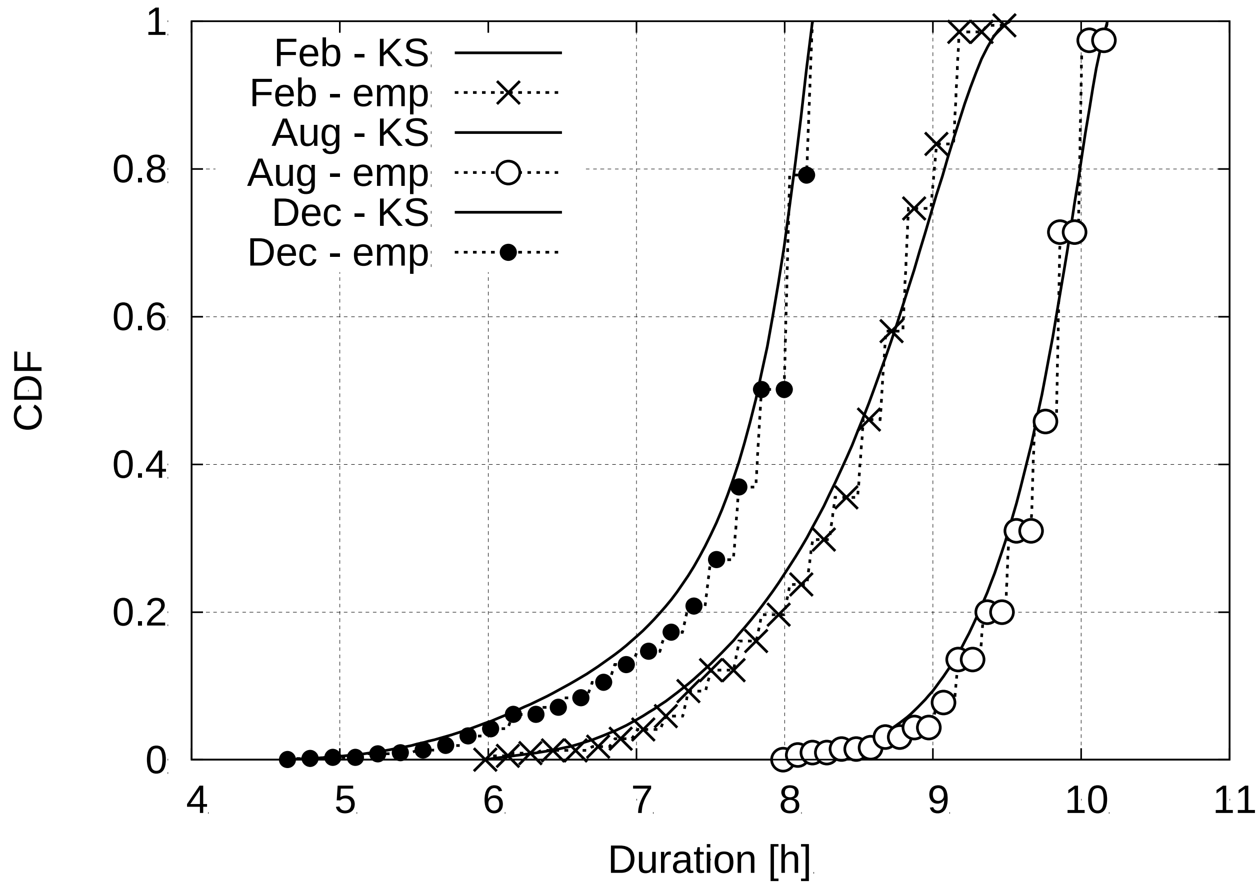
Statistics – State duration $f_d(x|x_s)$

- Los Angeles data collected from 1999 to 2010
- Solar module: square 6x6 cm^2
- Day state



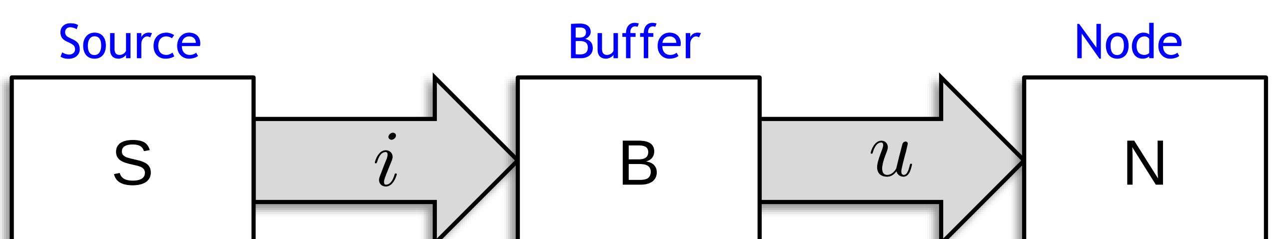
Statistics – State duration CDF

- Los Angeles data collected from 1999 to 2010
- Solar module: square 6x6 cm^2
- Day state



Stage

- Stage
  - time t during which the SMM remains in one state
- Energy income
  - r.v. drawn @ beginning of stage from $f_c(i|x_s)$
- Duration
  - r.v. drawn @ beginning of stage from $f_d(\tau|x_s)$
- Decision
  - u: chosen @ beginning of stage



Δ-charge (q) in stage k

- Balance between
  - Control u (current drained)
  - Input current i (from panel)
  - Stage duration τ

$$q = (i - u)\tau$$

- The pdf of the variation of charge in a stage is:

$$h(q|x_s, u) = \int_{-\infty}^{+\infty} \frac{1}{|\tau|} f_d(\tau|x_s) f_c(q/\tau + u|x_s) d\tau$$

Discrete Time Markov Chain (DTMC)

- Instead of using the SMM directly, we define an equivalent DTMC with slotted time
- Slot time is fixed
- When moving between stages (slots):
  - $h(q|x_s, u)$ models the d-charge variation
  - u is the control for the current stage
  - x_s is the source state in the current stage
  - q is the variation of charge in the battery
  - The new status of the battery is given by:

$$x_b(k) = \max\{0, \min\{x_b(k-1) + q, Q_{\max}\}\}$$

SYSTEM MODEL:
P1 – Optimal Operational Point

Energy Consumption Model

- MAC (channel access)
- Network topology
- Data gathering
- Networking
- Processing

$$I_X = i_X t_X f_X$$

Energy Consumption Model

- MAC (channel access)
- Network topology
- Data gathering
- Networking
- Processing

current drawn
spent in state X

$$I_X = i_X t_X f_X$$

Energy Consumption Model

- MAC (channel access)
- Network topology
- Data gathering
- Networking
- Processing

current drawn

average time spent in state X upon a transition to that state

$$I_X = i_X t_X f_X$$

Energy Consumption Model

- MAC (channel access)
- Network topology
- Data gathering
- Networking
- Processing

current drawn

average time spent

visits per second to state X (frequency)

$$I_X = i_X t_X f_X$$

Energy Consumption Model

- MAC (channel access)
- Network topology
- Data gathering
- Networking
- Processing

current drawn

average time spent

visits per second

average current drained

$$I_X = i_X t_X f_X$$

Energy Consumption Model

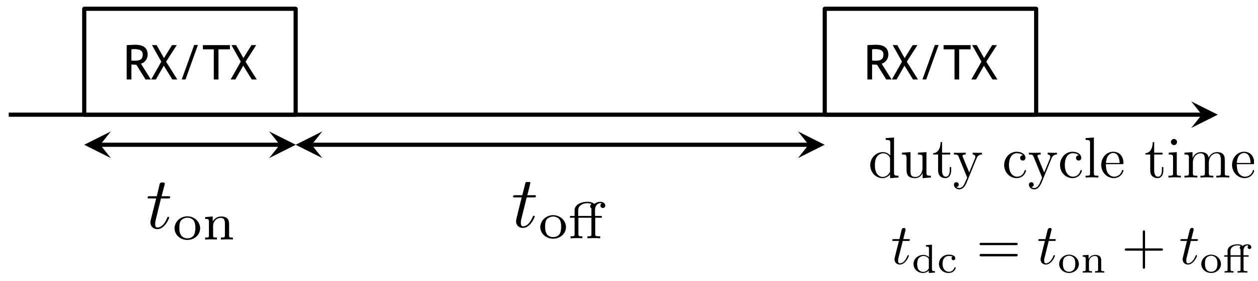
- Adding the contributions from all states

$$I_X = i_X t_X f_X$$ Average current drained in state X

$$r_X = t_X f_X$$ Fraction of time spent in state X

$$\begin{aligned} I_{\mathrm{OUT}} &= \sum_{i \in \mathcal{X}} I_i \\ &= I_{\mathrm{TX}} + I_{\mathrm{RX}} + I_{\mathrm{INT}} + I_{\mathrm{CPU}} + I_{\mathrm{IDLE-ON}} + I_{\mathrm{IDLE-OFF}} \end{aligned}$$

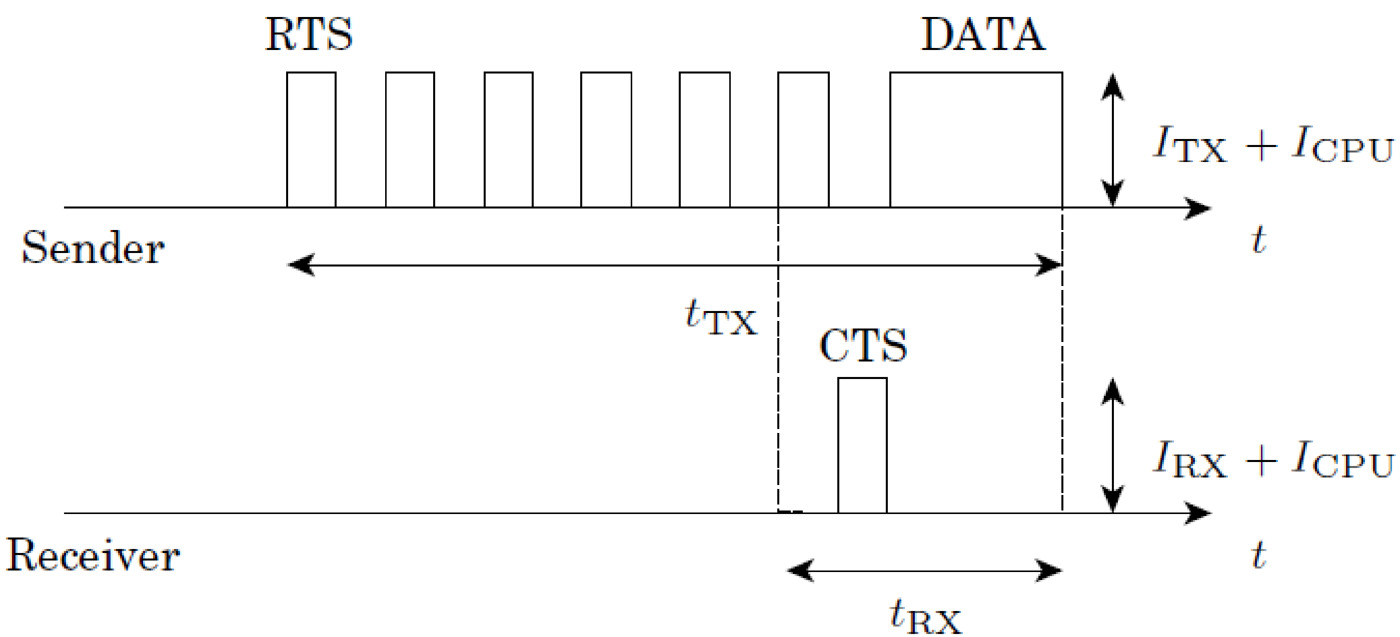
MAC: Duty-cycled WSN



- Energy consumption in TX, RX and IDLE is comparable
- Energy consumption is dominated by the PHY (radio)
- Commonly nodes are operated according to a duty-cycled approach
- The duty cycle d [%] is defined as: $$d = \frac{t_{\mathrm{on}}}{t_{\mathrm{on}} + t_{\mathrm{off}}}$$
- d usually ranges between 0.01 □ 0.1 (nodes are awake 1 to 10% of the time)

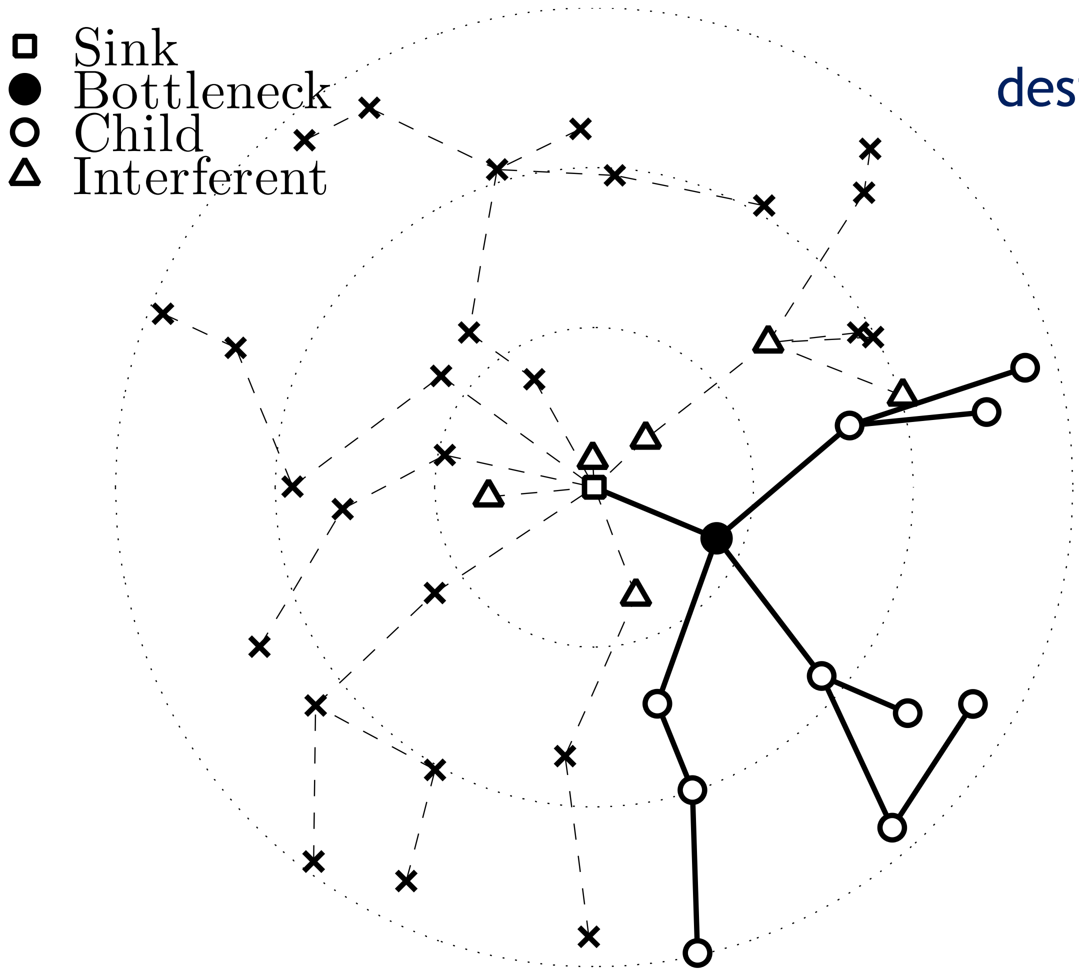
MAC: Access Protocol

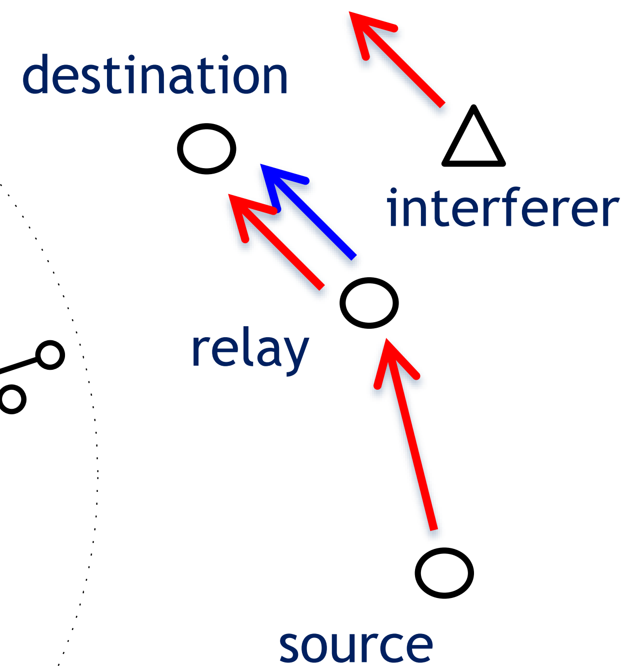
- X-MAC is used as the channel access technique
- allows asynchronous communication
- preamble-based, transmitter initiated



$$t_{\mathrm{TX}} = t_{\mathrm{on}} + t_{\mathrm{off}}/2 + t_{\mathrm{data}} + (f'_{\mathrm{U}}/f_{\mathrm{U}} - 1)t_{\mathrm{dc}}$$

Topology, Routing & Bottleneck





Topology, Routing & Bottleneck

Bottleneck node is the sensor that, due to its position in the network, is the fastest to deplete its battery

n_{c} number of children nodes, i.e., total number of nodes in the sub-tree rooted at the bottleneck

n_{i} number of interfering nodes (within the transmission range of the bottleneck)

n_{int} accumulated number of interfering packets from interfering nodes, accounting for endogenous (their own transmissions) and exogenous (transmissions of their children nodes) traffic

Data Gathering and Networking

- Data gathering requires all nodes to periodically send information to the sink
- Networking implies a bi-directional communication among all nodes

$$f_{\mathrm{TX,DG}} = (1 + n_{\mathrm{c}})/t_{\mathrm{U}}$$
$$f_{\mathrm{RX,DG}} = n_{\mathrm{c}}/t_{\mathrm{U}}$$
$$f_{\mathrm{TX,RPL}} = (2 + n_{\mathrm{c}})/t_{\mathrm{rpl}}$$
$$f_{\mathrm{RX,RPL}} = (1 + n_{\mathrm{i}} + n_{\mathrm{c}})/t_{\mathrm{rpl}}$$

Optimization Problem P1

$t_{\mathrm{U}} \rightarrow$ average inter-packet transmission time

$t_{\mathrm{dc}} \rightarrow$ duty cycle period $(t_{\mathrm{dc}} = t_{\mathrm{on}} + t_{\mathrm{off}})$

$I_{\mathrm{OUT}} \rightarrow$ average current drained

$u \rightarrow$ maximum current drained

$(u \rightarrow$ control action set by P2)

Optimization Problem P1

$$\begin{aligned} \underset{t_{\mathrm{U}},t_{\mathrm{dc}}}{\text{maximize}}\ & f_{\mathrm{U}} \\ \text{subject to: } & I_{\mathrm{OUT}} \le u, \\ & r_x \ge 0,\ \forall x \in \chi_N, \\ & t_{\mathrm{U}} \ge 0,\ t_{\mathrm{dc}} \ge t_{\mathrm{on}}. \end{aligned}$$

- Constraints are polynomials
  - Convex optimization is possible
- Closed-form solution is available

Optimization Problem P1

- Problem setup:
  - Hardware defines consumption figures
  - MAC defines timings
  - Routing and topology define relays and bottleneck

- Result $(t^*_{\mathrm{U}}, t^*_{\mathrm{dc}})$
  - Data gathering time and Duty-cycle
  - Achieving the maximum throughput
  - Subject to the given consumption u

Example: Parameters

Timings

$$t_{\text{on}} = 6 \text{ ms}$$

$$t_{\text{off}} = 14 \text{ ms}$$

$$t_{\text{int}} = 10 \text{ ms}$$

$$t_{\text{CPU}} = 40 \text{ ms}$$

$$t_{\text{RPL}} = 6 \text{ h}$$

Energy

$$i_{\text{TX}} = 10 \text{ mA}$$

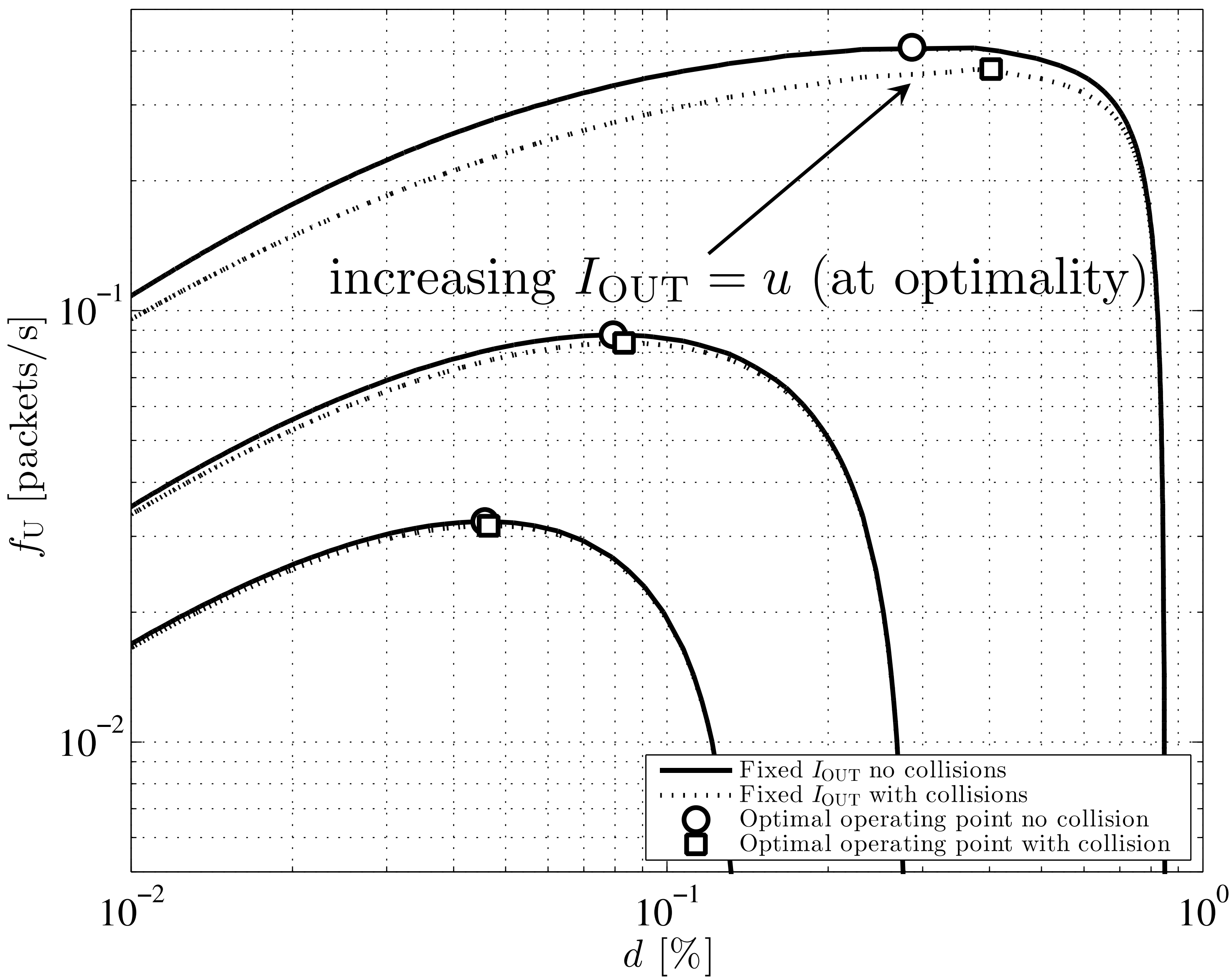
$$i_{\text{RX}} = 8.7 \text{ mA}$$

$$i_{\text{CPU}} = 26.6 \text{ mA}$$

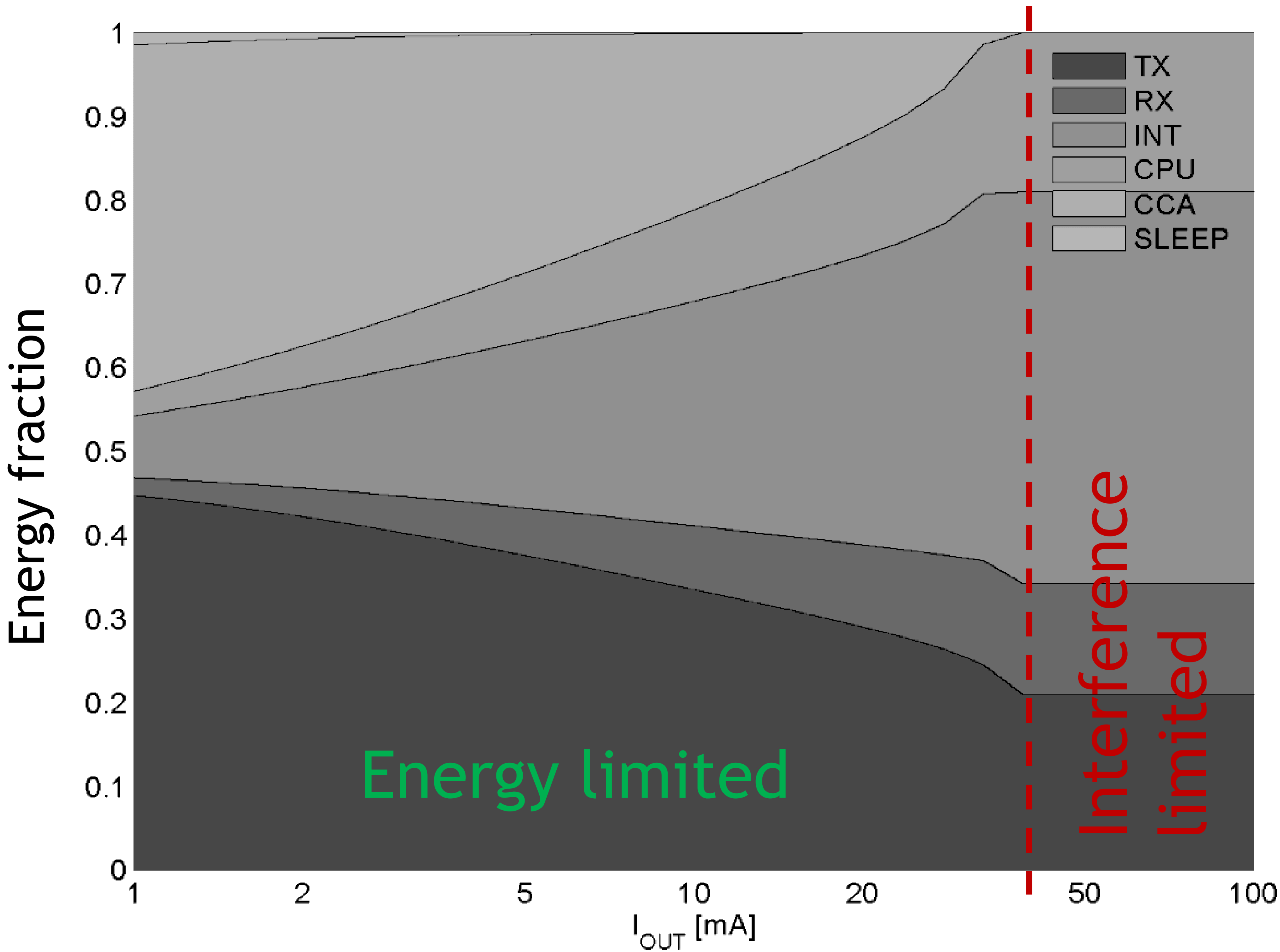
$$i_{\text{S}} = 0.015 \text{ mA}$$

$$K_{\text{U}} = 10$$

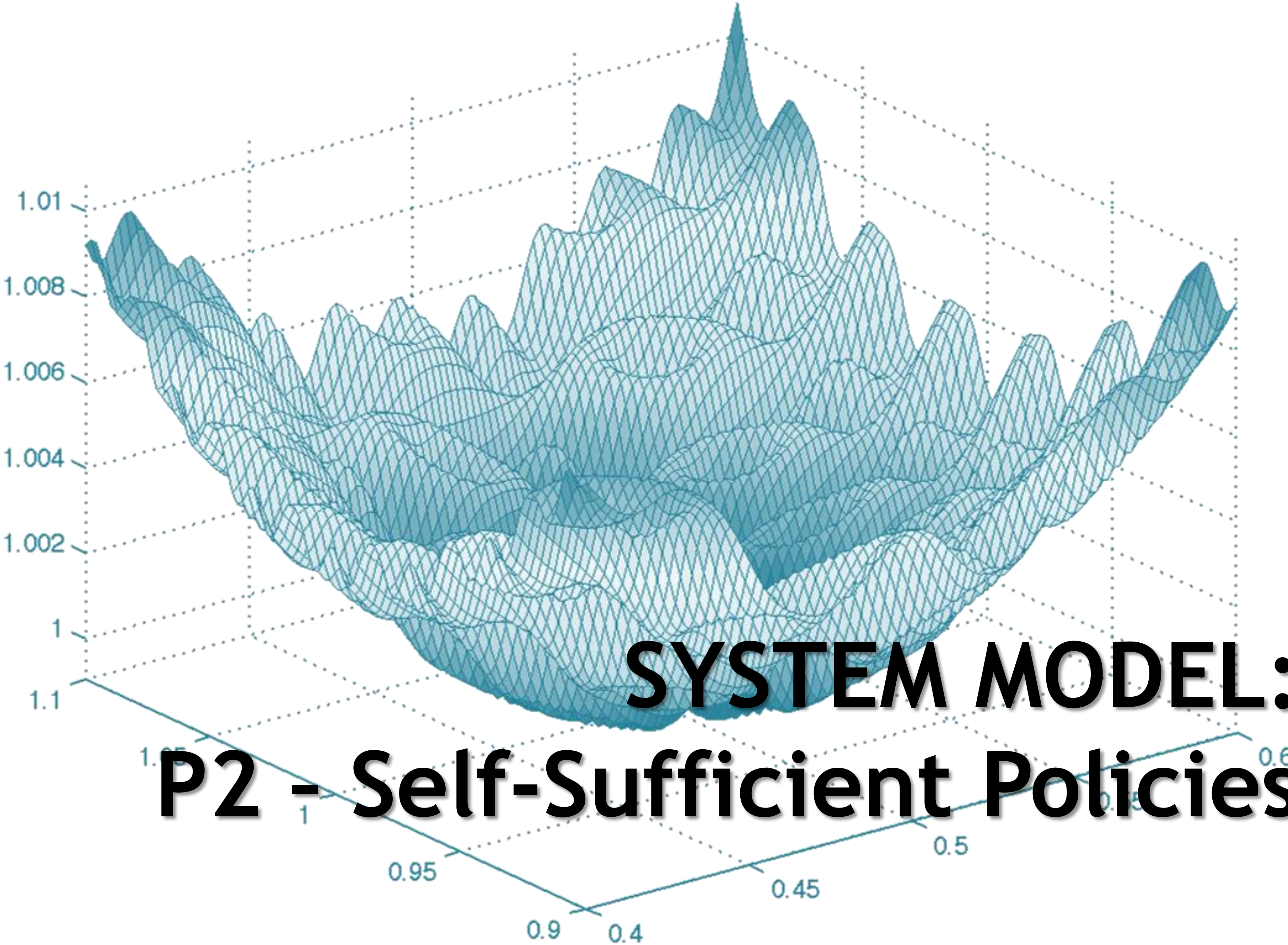
Example: Solution



Example: Energy State Breakdown



SYSTEM MODEL:
P2 – Self-Sufficient Policies

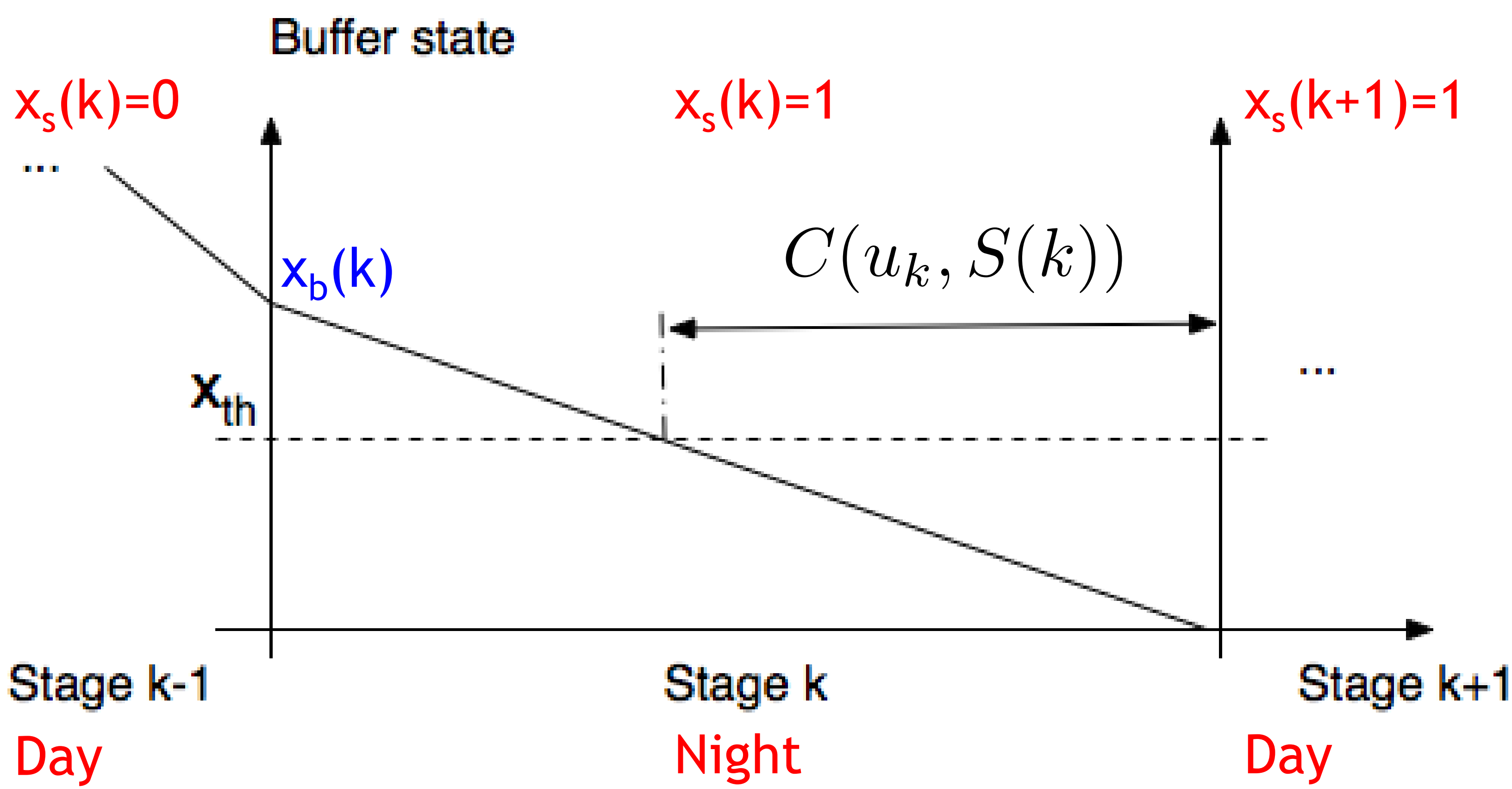


Problem Formulation

- **System state** $S(k)=(x_b(k),x_s(k))$
  - Energy buffer state $x_b(k)$ @ beginning of stage k
  - Energy source state $x_s(k)$ during stage k
- **Control** u_k
  - Current drained by the node (I_{out})
  - Immediate reward $R(u_k,S(k))$ (throughput [pkt/s])
- **Stage-Cost** $C(u_k,S(k))$
  - Average time spent with $x_b(k)<x_{th}(k)$ (outage threshold)
- **Stage-Reward** $R(u_k,S(k))$
  - Total reward: integral of $R(u_k,S(k))$ over the solution path

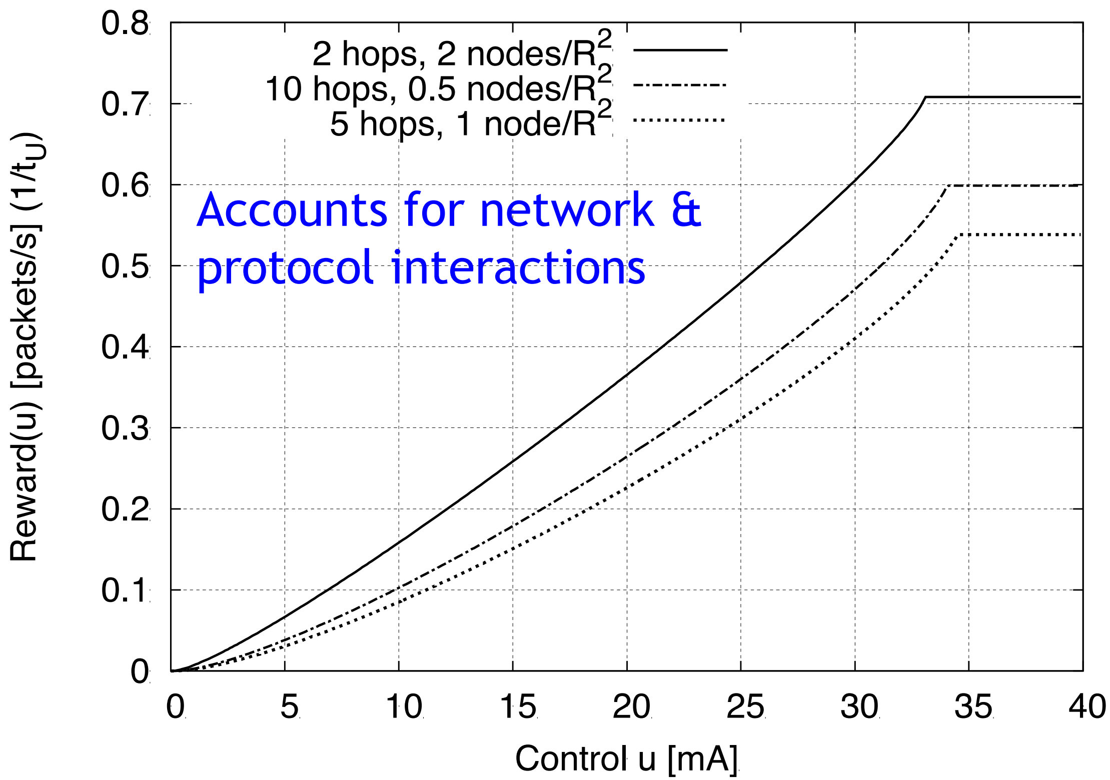
Single Stage Cost

- Average single-stage cost
  - Duration of time interval where $x_b(k) < x_{th}$



Single Stage Reward

- Average reward over a single stage k:
  - $R(u_k, S_k)$ multipled by the time during which $x_b>0$ (non-empty buffer)
  - $R(u_k, S_k)$ is obtained from problem P1



Optimal Policies

- Find the policy π: $(u_k(S(k))$, that maximizes the expected long-term throughput (reward)
- Long-term expected cost below a threshold (fraction of time with low buffer state)
- Discount factor α: control the look-ahead capability of the optimal policy

$$\underset{\pi}{\text{maximize}} \left\{ \lim_{N \to +\infty} E \left[\sum_{k=0}^{N-1} \alpha^k R(u_k, S(k)) \middle| S(0) \right] \right\}$$

$$\text{subject to: } \lim_{N \to +\infty} E \left[\sum_{k=0}^{N-1} \alpha^k C(u_k, S(k)) \middle| S(0) \right] \leq C_{\text{th}}$$

Lagrangian Reward (1)

- A lagrangian λ is introduced to balance
  - Costs $C(u_k,S(k))$
  - Rewards $R(u_k,S(k))$
- The lagrangian is part of the solution
  - Choose λ
  - Solve optimal problem for this λ
  - Iterate over λ to find global optimum

$$R_{\mathcal{L}}(u_k, S(k)) = \underline{R(u_k, S(k))} - \lambda \underline{C(u_k, S(k))}$$

Lagrangian Reward (2)

- Large λ: cost prevails
  - Small reward, Small cost
    - cost constraint is satisfied whp
    - λ can be decreased
- Small λ: reward prevails
  - High reward (more aggressive policies), large cost
    - cost constraint is not satisfied
    - λ has to be increased
- Binary search over λ

$$R_{\mathcal{L}}(u_k, S(k)) = \underline{R(u_k, S(k))} - \lambda \underline{C(u_k, S(k))}$$

Lagrangian Bellman Equation

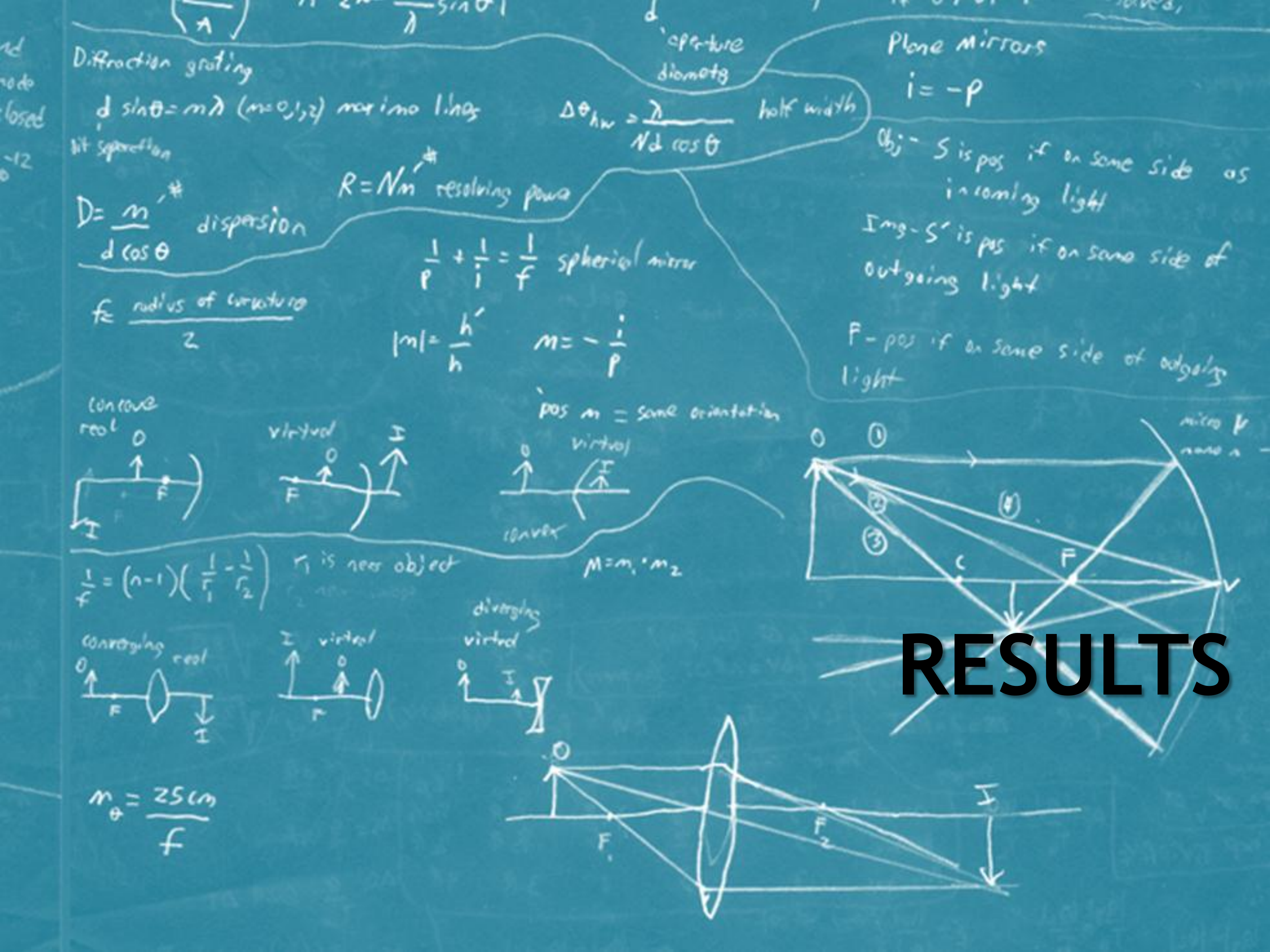
- Expected reward from stake k onwards

$$J(S(k)) = \max_{u_k \in U(S(k))} \left\{ E[R_{\mathcal{L}}(u_k, S(k))] + \alpha \int_{-\infty}^{+\infty} J(S'(k+1)) h(q|x_s(k), u_k) \right\}$$

  - Single-stage expected Lagrangian reward
  - Discounted future expected reward weighted by the distribution of the charge variation

- Next stage

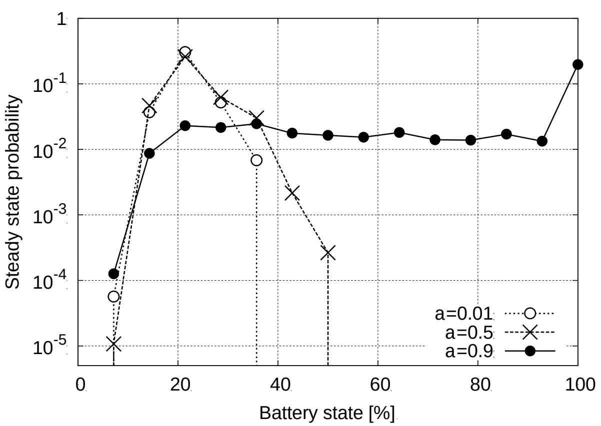
$$\begin{aligned} S'(k+1) &= (x_s(k+1), x_b(k+1)) \\ &= (1 - x_s(k), \max\{\min\{q + x_b, Q\}, 0\}) \end{aligned}$$

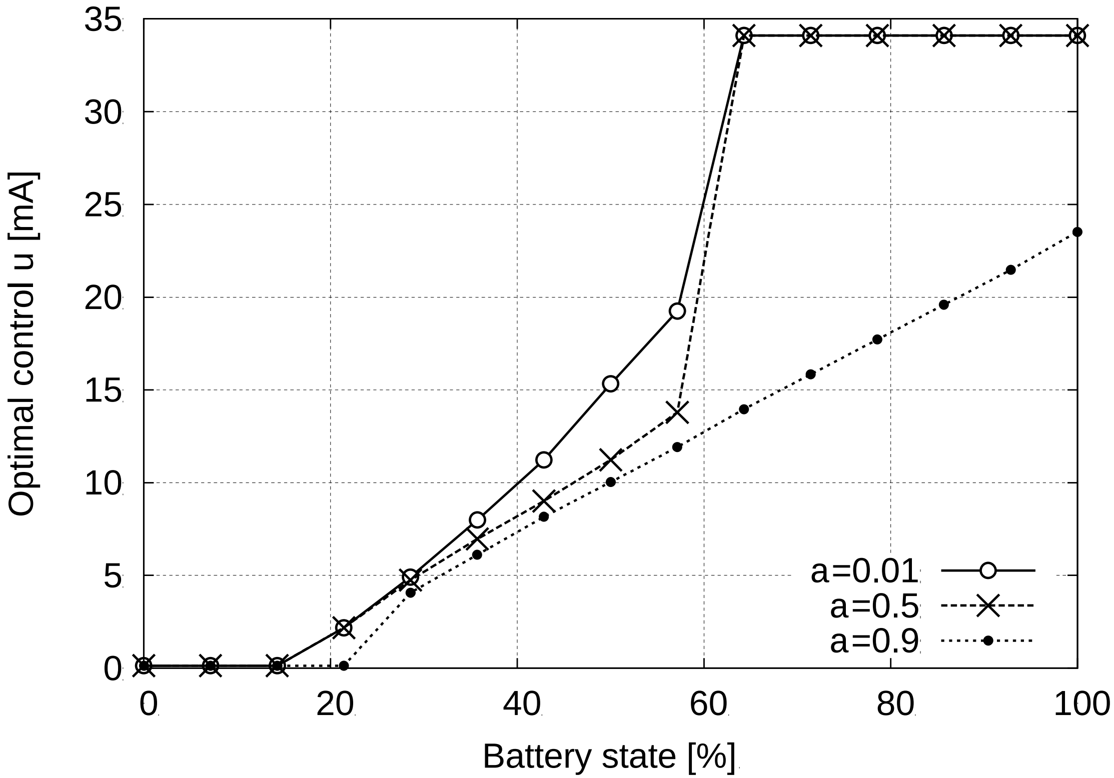

Diffraction grating
dispersion
resolving power
half width
Plane mirrors
spherical mirror
incoming light
outgoing light
pos m = same orientation
converging
diverging
virtual
real
concave
convex

RESULTS

Discount Factor

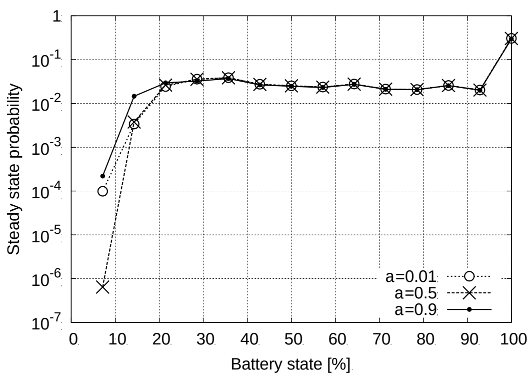
- A larger discount factor increases the optimization horizon and makes the policies more conservative
- Day

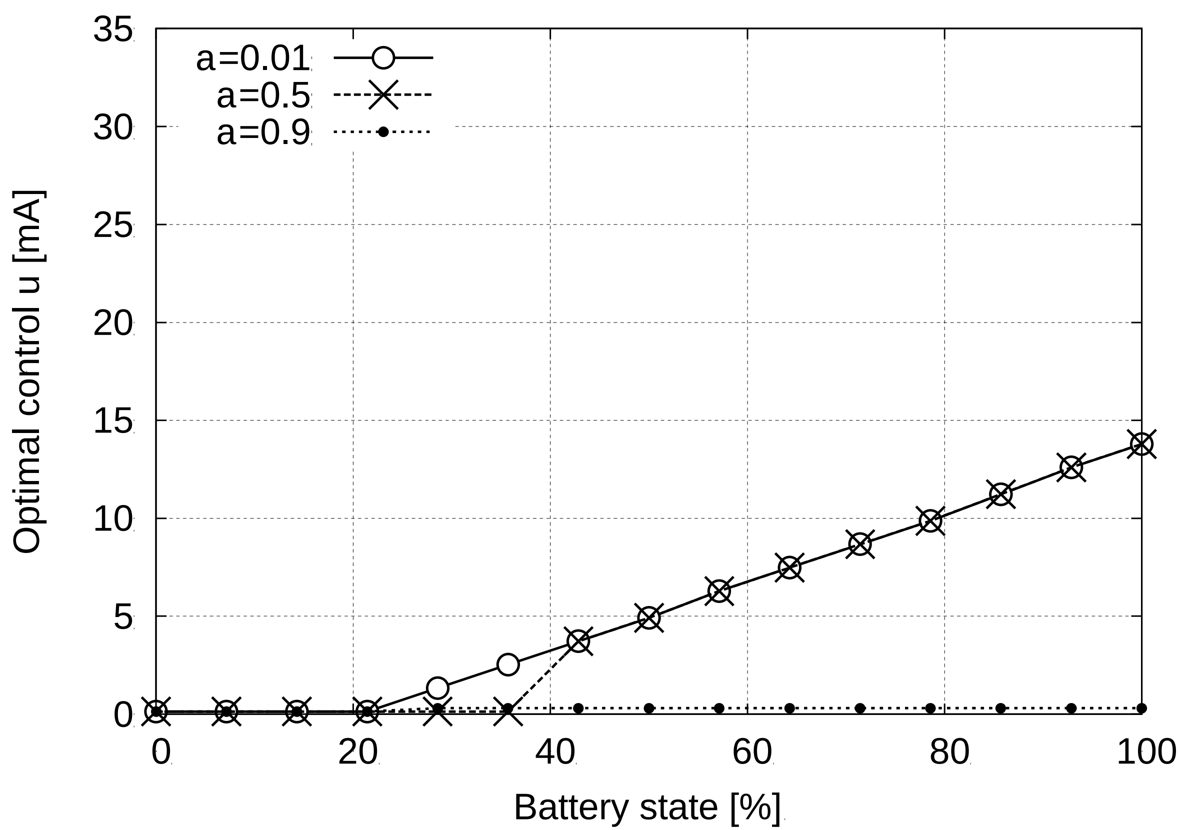




Discount Factor

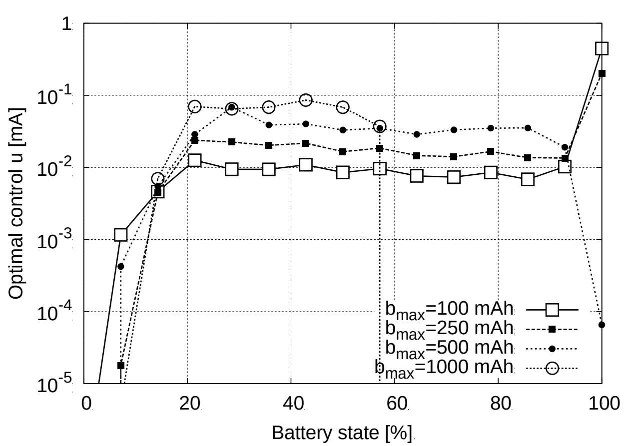
- A larger discount factor increases the optimization horizon and makes the policies more conservative
- Night

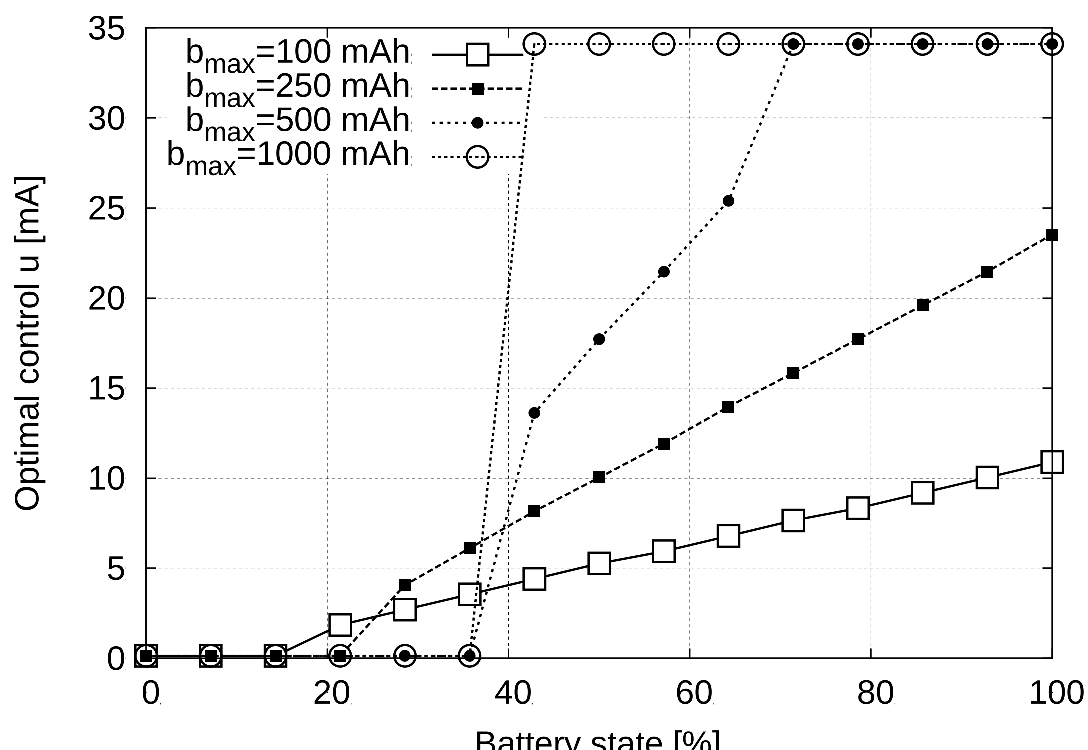




Battery Size

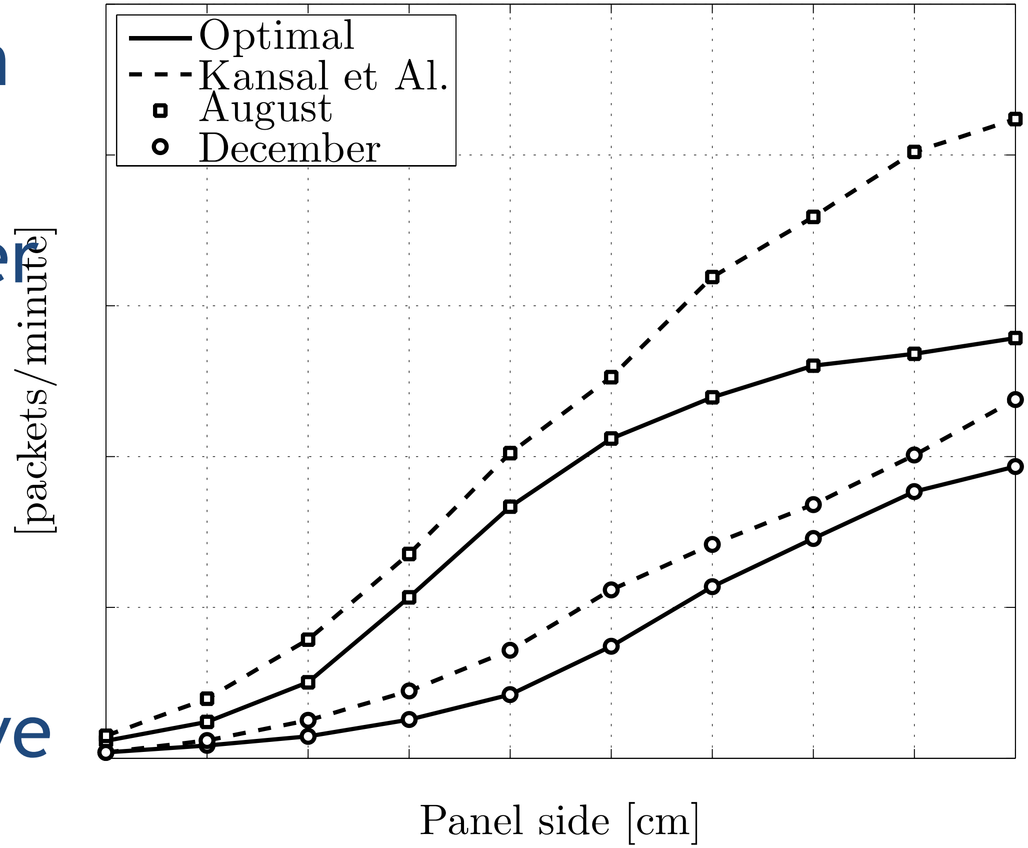
- Larger batteries allows for more aggressive policies with threshold behavior





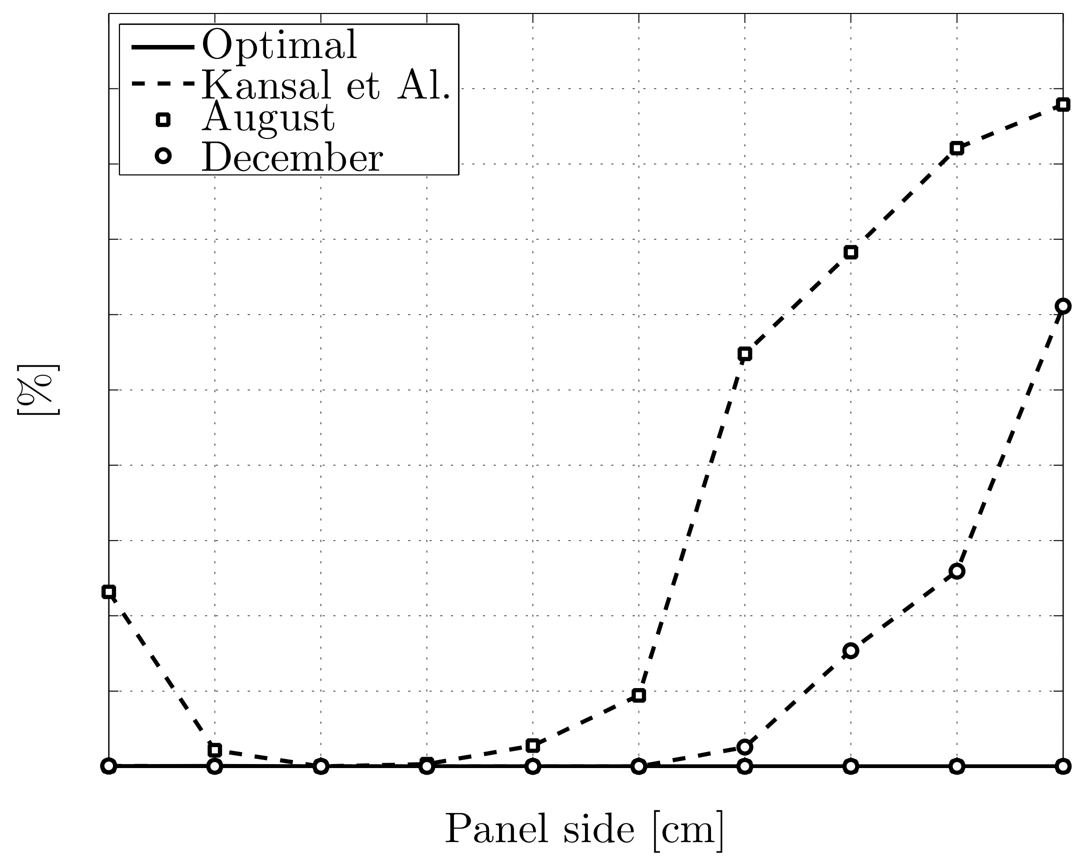
Throughput vs Panel Size

- Kansal et Al. propose a predictive approach that allows for higher throughput, but the aggressiveness is payed with higher outage rate
- Our optimization is bound not to violate the buffer condition
- More conservative

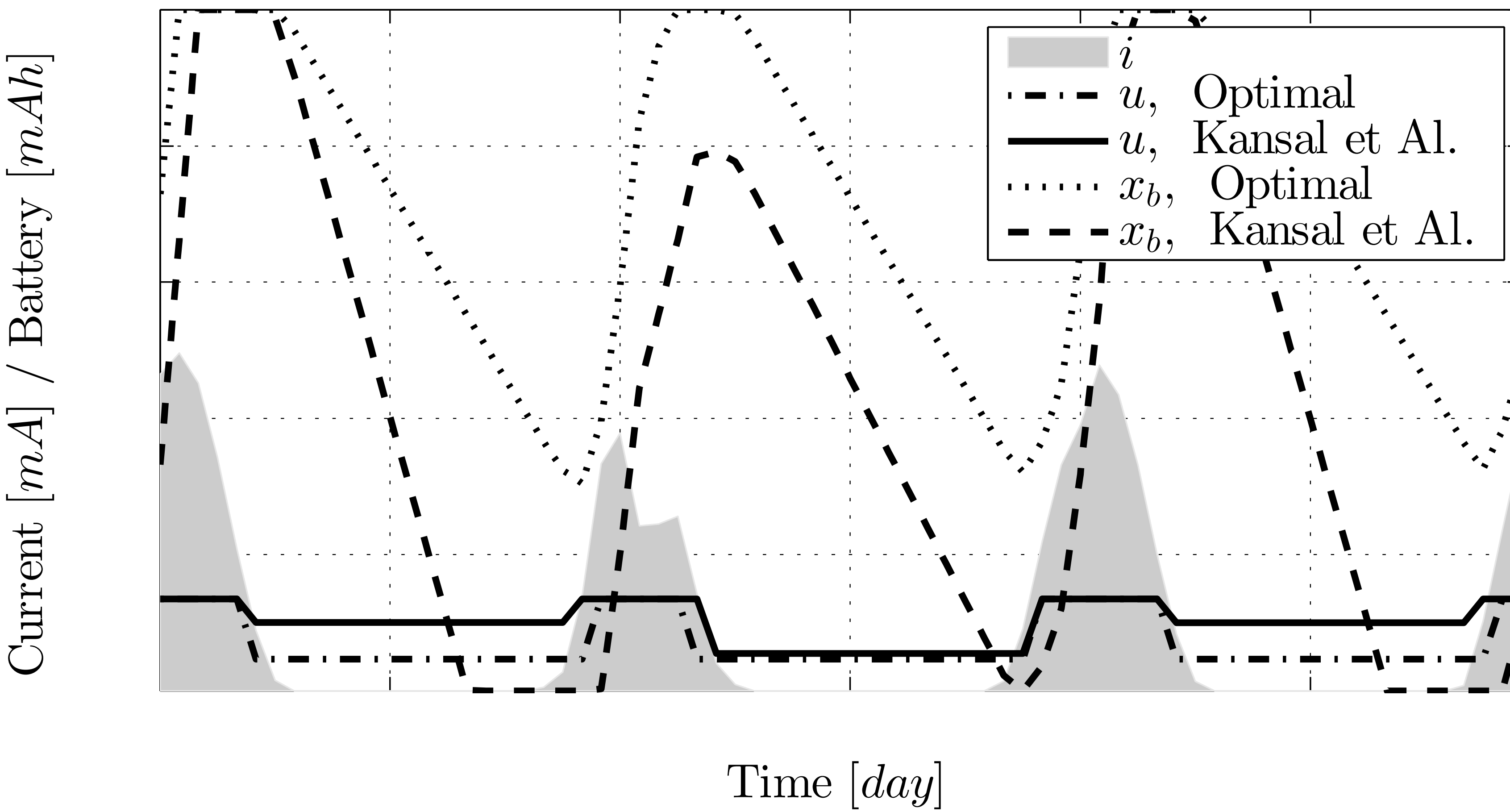


Outage vs Panel Size

- The more aggressive approach cannot prevent battery outage, which instead is avoided by our policies



Time variation



Bibliography

[1] N. Bui and M. Rossi, "Staying Alive: System Design for Self-Sufficient Sensor Networks," *ACM Transactions on Sensor Networks*, Vol.11, No 3, May 2015.

[2] M. Miozzo, D. Zordan, P. Dini and M. Rossi, "SolarStat: Modeling Photovoltaic Sources through Stochastic Markov Sources," IEEE *EnergyCon*, Dubvronik, Croatia, May 2014.

[3] N. Bui and M. Rossi, "Dimensioning Self-Sufficient Networks of Energy Harvesting Embedded Devices," *WifFex*, Kaliningrad, Russia, September 2013.

[4] A. Kansal, J. Hsu, S. Zahedi and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 6, No 4, 2007.

THANKS FOR THE ATTENTION

Self-sufficient Sensor Networks: from Energy Profiling to Optimal Sustainable Performance

Nicola Bui, nicola.bui@imdea.org
(a joint work with Michele Rossi, rossi@dei.unipd.it)

ANY QUESTION?

institute imdea networks

Developing the Science of Networks

BACKUP SLIDES

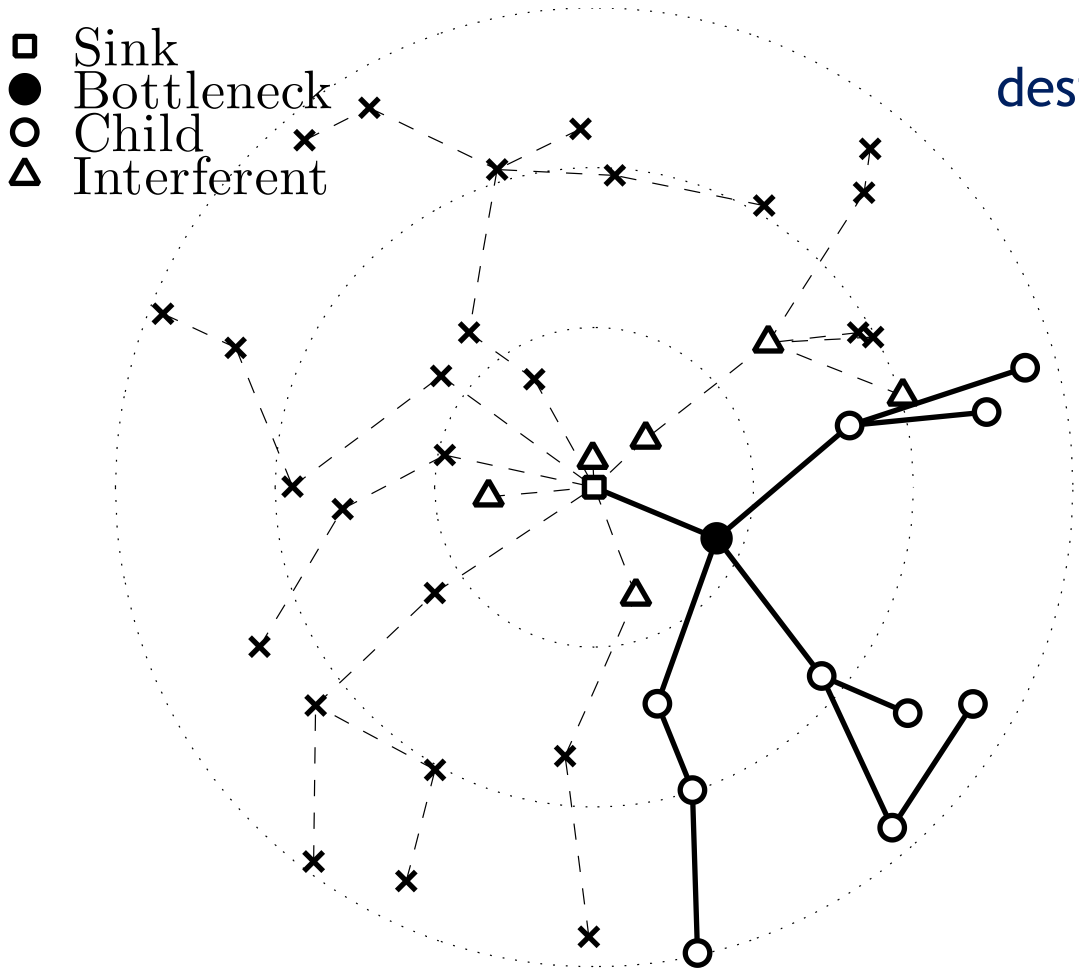
DISCUSSION

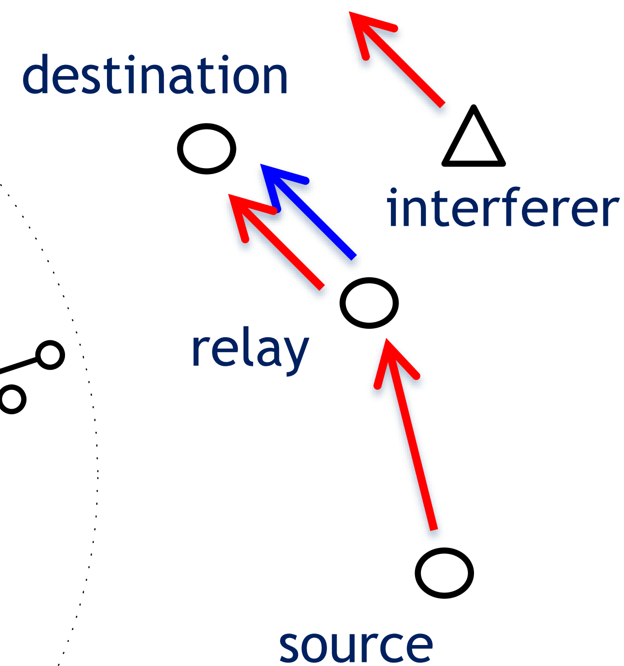
Variation

- LTE Discontinuous reception
  - Sleep time optimization against application latency
- D2D Communication
  - Terminals may relay traffic from other terminals to the base station

P1: DETAILS

Topology, Routing & Bottleneck





Topology, Routing & Bottleneck

Bottleneck node is the sensor that, due to its position in the network, is the fastest to deplete its battery

n_{c} number of children nodes, i.e., total number of nodes in the sub-tree rooted at the bottleneck

n_{i} number of interfering nodes (within the transmission range of the bottleneck)

n_{int} accumulated number of interfering packets from interfering nodes, accounting for endogenous (their own transmissions) and exogenous (transmissions of their children nodes) traffic

Problem 1: DATA TX, RX

Transmitted and received packets per second due to Data Gathering (DG)

$$f_{\mathrm{TX,DG}} = (1 + n_{\mathrm{c}})/t_{\mathrm{U}}$$

$$f_{\mathrm{RX,DG}} = n_{\mathrm{c}}/t_{\mathrm{U}}$$

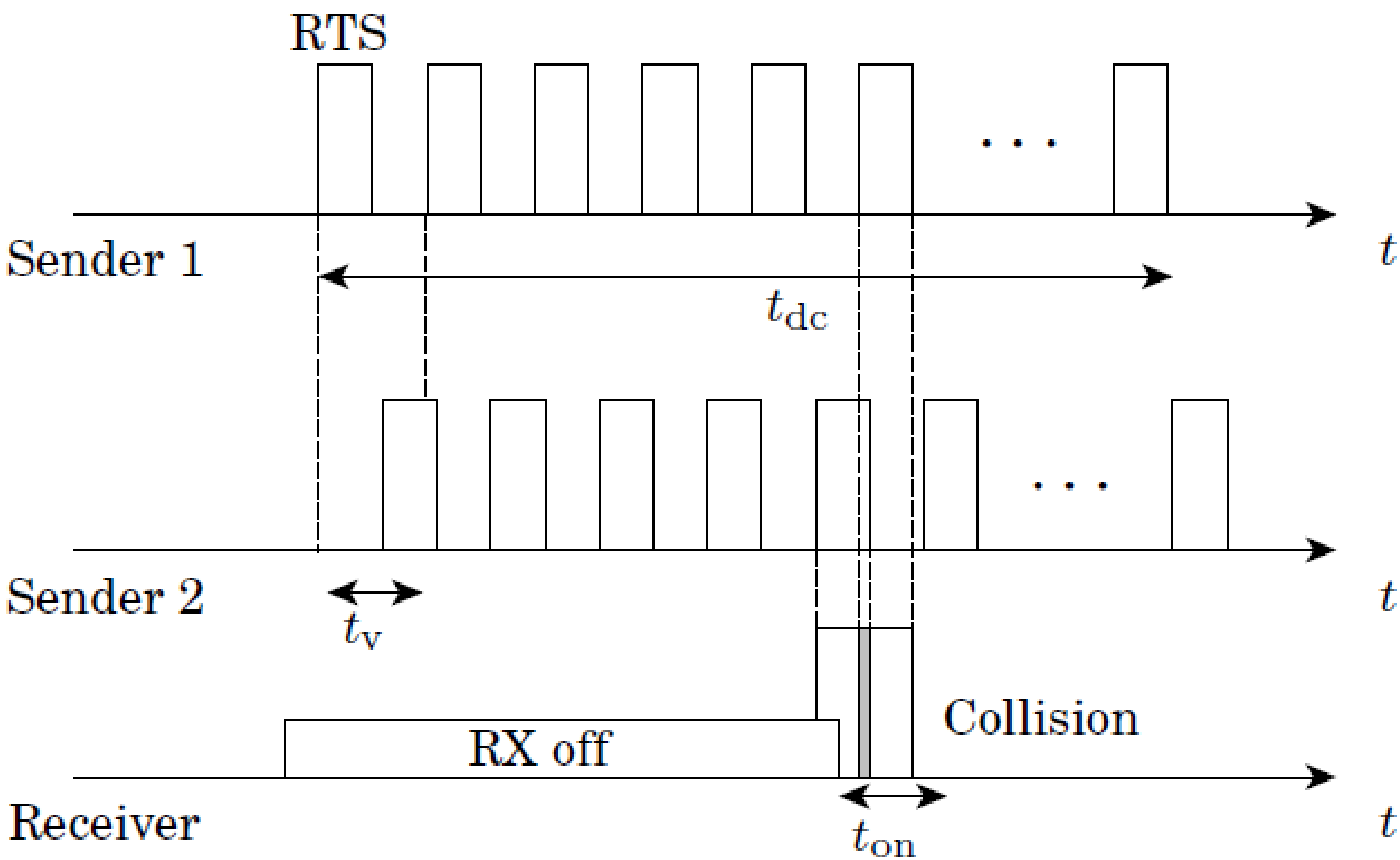
Packet transmission time (including collisions)

$$t_{\mathrm{TX}} = t_{\mathrm{on}} + t_{\mathrm{off}}/2 + t_{\mathrm{data}} + (f'_{\mathrm{U}}/f_{\mathrm{U}} - 1)t_{\mathrm{dc}}$$

f'_{U} : TX frequency with collisions

f_{U} : TX frequency wo collisions

Problem 1: Collisions



Problem 1: RPL & DODAG

- Destination Oriented Directed Acyclic Graph (DODAG)
- RPL defines new ICMPv6 messages:

- Dag Information Object (DIO): carries information that allows nodes to discover an DODAG instance, learn its config pars and select a parent node
- Destination Advertisement Object (DAO): used to propagate destination information upwards the DODAG
- Dag Information Solicitation (DIS): to solicitate the TX of a DODAG object from an RPL node (not used in the analysis)

Problem 1: DODAG upward routes

- Nodes periodically send link-local (broadcast) DIO messages
- Nodes listen for DIOs and use the information therein to construct a DODAG or maintain an existing one
- Based on the info on the DIOs a node chooses its parent so as to minimize the cost toward the DODAG root
- Analysis: DIOs are periodically sent by the nodes at a rate $1/t_{\mathrm{rpl}}$

Dept. of Mathematics, UNIPD

Problem 1: DODAG downward routes

- Nodes inform parents of their present and reachability to their descendants by sending a DAO message
- DAOs are aggregated at intermediate nodes while sent upstream
- DAOs propagate from the leaves to the DODAG root node
- Analysis: DAOs are sent by the leaf nodes at a rate $1/t_{\mathrm{rpl}}$

Dept. of Mathematics, UNIPD

Problem 1: RPL TX, RX

- 1 DIO and 1 DAO msg from bottleneck, n_c DAOs from its children nodes:

$$f_{\mathrm{TX,RPL}} = (2 + n_{\mathrm{c}})/t_{\mathrm{rpl}}$$

- 1 DIO from parent, n_c DAOs from children nodes, n_i DIOs from inter. nodes:

$$f_{\mathrm{RX,RPL}} = (1 + n_{\mathrm{i}} + n_{\mathrm{c}})/t_{\mathrm{rpl}}$$

- DIOs are not treated as interference as they are broadcast (*ergo* they are received and treated as legitimate packets)

Problem 1: interference

- Rate of interfering packets:

$$f_{\mathrm{INT}} = n_{\mathrm{int}}(1/t_{\mathrm{U}} + 1/t_{\mathrm{rpl}})$$

- TX rate for data packets

$$1/t_{\mathrm{U}}$$

- TX rate for RPL DODAG control packets

$$1/t_{\mathrm{rpl}}$$

Problem 1: current consumption figures

TX time for a single pkt transmission

$$I_{\mathrm{TX}} = (i_{\mathrm{c}} + i_{\mathrm{t}})\boxed{[t_{\mathrm{dc}}/2 + t_{\mathrm{on}}/2 + t_{\mathrm{data}} + (f'_{\mathrm{U}}/f_{\mathrm{U}} - 1)t_{\mathrm{dc}}]} \times$$

$$\times \boxed{[(1 + n_{\mathrm{c}})/t_{\mathrm{U}} + (2 + n_{\mathrm{c}})/t_{\mathrm{rpl}}]}$$

transmissions rate [pps]

$$I_{\mathrm{RX}} = (i_{\mathrm{c}} + i_{\mathrm{r}})t_{\mathrm{data}}[n_{\mathrm{c}}/t_{\mathrm{U}} + (1 + n_{\mathrm{c}} + n_{\mathrm{i}})/t_{\mathrm{rpl}}]$$

$$I_{\mathrm{INT}} = (i_{\mathrm{c}} + i_{\mathrm{r}})t_{\mathrm{int}}n_{\mathrm{int}}(1/t_{\mathrm{U}} + 1/t_{\mathrm{rpl}})$$

$$I_{\mathrm{CPU}} = i_{\mathrm{c}}t_{\mathrm{cpu}}K_{\mathrm{U}}/t_{\mathrm{U}}$$

$$I_{\mathrm{CCA}} = (i_{\mathrm{c}} + i_{\mathrm{r}})d_{\mathrm{c}}r_{\mathrm{IDLE}}$$

$$I_{\mathrm{OFF}} = i_{\mathrm{s}}(1 - d_{\mathrm{c}})r_{\mathrm{IDLE}}$$

Dept. of Mathematics, UNIPD

Problem 1: current consumption figures

$$I_{\mathrm{TX}} = (i_{\mathrm{c}} + i_{\mathrm{t}})[t_{\mathrm{dc}}/2 + t_{\mathrm{on}}/2 + t_{\mathrm{data}} + (f'_{\mathrm{U}}/f_{\mathrm{U}} - 1)t_{\mathrm{dc}}] \times$$
$$\times\ [(1 + n_{\mathrm{c}})/t_{\mathrm{U}} + (2 + n_{\mathrm{c}})/t_{\mathrm{rpl}}]$$

$$I_{\mathrm{RX}} = (i_{\mathrm{c}} + i_{\mathrm{r}})t_{\mathrm{data}}[n_{\mathrm{c}}/t_{\mathrm{U}} + (1 + n_{\mathrm{c}} + n_{\mathrm{i}})/t_{\mathrm{rpl}}]$$

$$I_{\mathrm{INT}} = (i_{\mathrm{c}} + i_{\mathrm{r}})t_{\mathrm{int}}n_{\mathrm{int}}(1/t_{\mathrm{U}} + 1/t_{\mathrm{rpl}})$$

$$I_{\mathrm{CPU}} = i_{\mathrm{c}}t_{\mathrm{cpu}}K_{\mathrm{U}}/t_{\mathrm{U}} \Rightarrow$$ CPU time due to pkt generation

$$I_{\mathrm{CCA}} = (i_{\mathrm{c}} + i_{\mathrm{r}})d_{\mathrm{c}}r_{\mathrm{IDLE}} \Rightarrow$$ CPU time due to IDLING - RADIO ON

$$I_{\mathrm{OFF}} = i_{\mathrm{s}}(1 - d_{\mathrm{c}})r_{\mathrm{IDLE}} \Rightarrow$$ CPU time due to IDLING - RADIO OFF

$$r_{\mathrm{IDLE}} = 1 - r_{\mathrm{TX}} - r_{\mathrm{RX}} - r_{\mathrm{INT}} - r_{\mathrm{CPU}}$$

Dept. of Mathematics, UNIPD

Problem 1: closed-form solution

1) $$I_{\mathrm{OUT}}(t_{\mathrm{U}}, t_{\mathrm{dc}}) = \sum_{i \in \mathcal{X}} I_i$$ Total power consumption

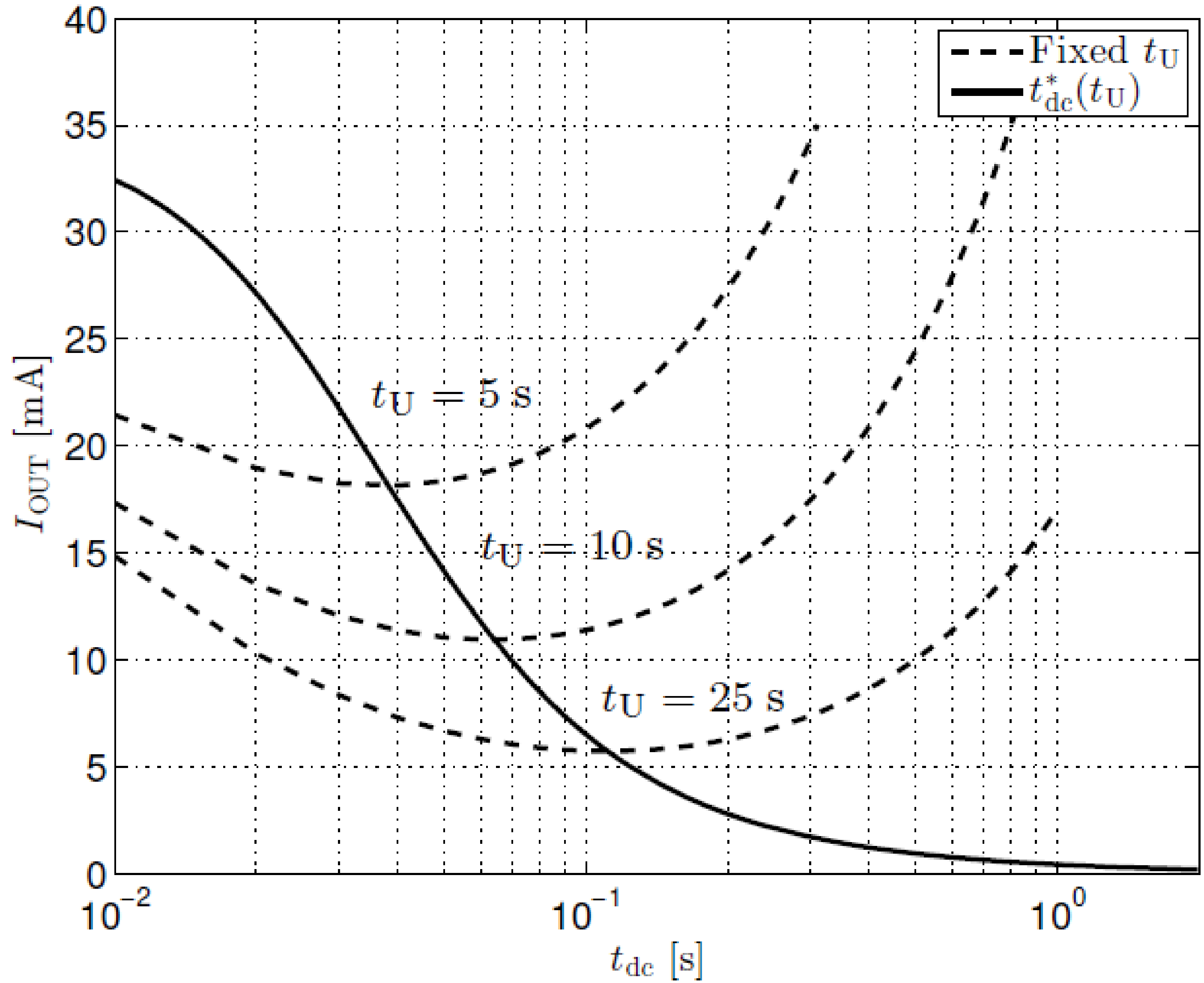
2) $$\frac{\partial I_{\mathrm{OUT}}(t_{\mathrm{U}}, t_{\mathrm{dc}})}{\partial t_{\mathrm{dc}}} = 0 \rightarrow t^*_{\mathrm{dc}}(t_{\mathrm{U}})$$ Optimal duty-cycle

3) $$I_{\mathrm{OUT}}(t_{\mathrm{U}}, t^*_{\mathrm{dc}}(t_{\mathrm{U}})) - u = 0 \rightarrow t^*_{\mathrm{U}}(u)$$

$$u \in [u_{\min}, u_{\max}]$$

$t^*_{\mathrm{U}}(u)$ Min. inter-packet TX time for given u

Problem 1: optimal duty-cycle



TECHNICAL SOLUTIONS

Heuristic (1/2)

Heterogeneous buffer states

- DAOs are used to periodically report data (status of the nodes, etc.) to the DODAG root (i.e., the sink)
- We use these messages to periodically collect the energy buffer status of all nodes
- The sink decides which policy to adopt based on the minimum among all buffer states, $\min_i (B_i)$

Dept. of Mathematics, UNIPD

Heuristic (2/2)

- The optimal policy is computed for the bottleneck node (worst case network parameters)
- This policy is used to decide the maximum energy consumption level for all nodes…
- …based on the minimum among all buffer states, $\min_i (B_i)$

Outcome

- Policy will be suboptimal
- But will assure energy sustainability at all nodes

Dept. of Mathematics, UNIPD

Heuristic – Results (1/2)

- BN: bottleneck node
- SBN: second-bottleneck node
  - Located in the sub-tree originating from the BN
  - With the second-highest energy consumption
- Worst case assumption
  - The SBN has the same parameters n_i, n_{int} as the BN
  - As just one node less as its number of children, i.e., n_c-1

Heuristic – Results (2/2)