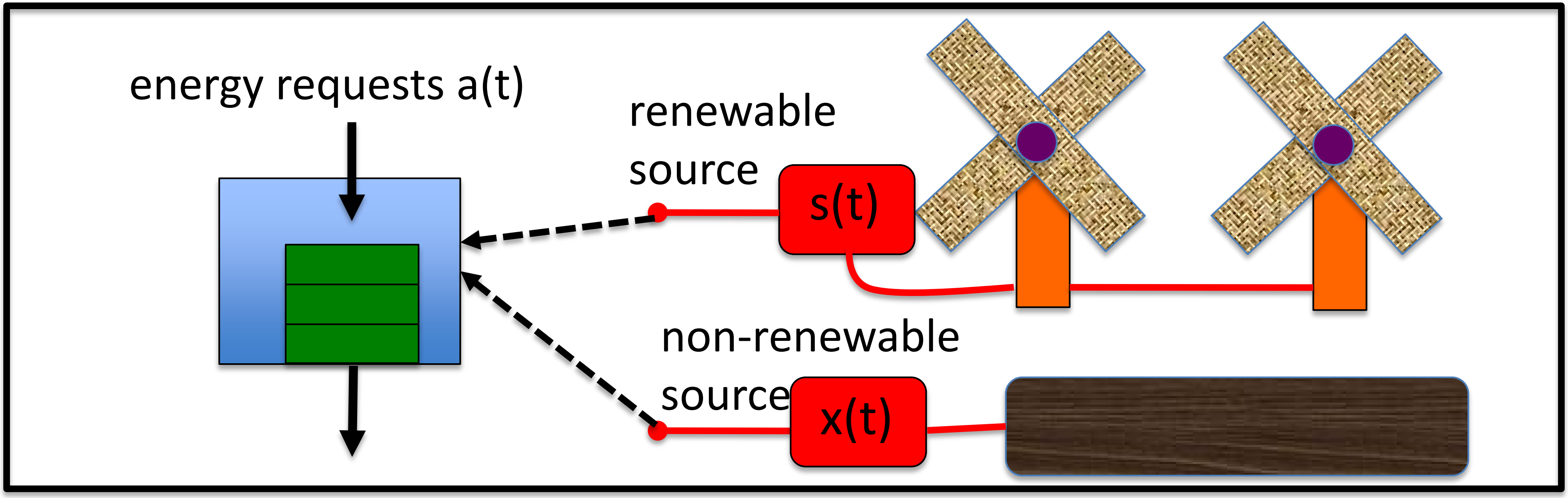
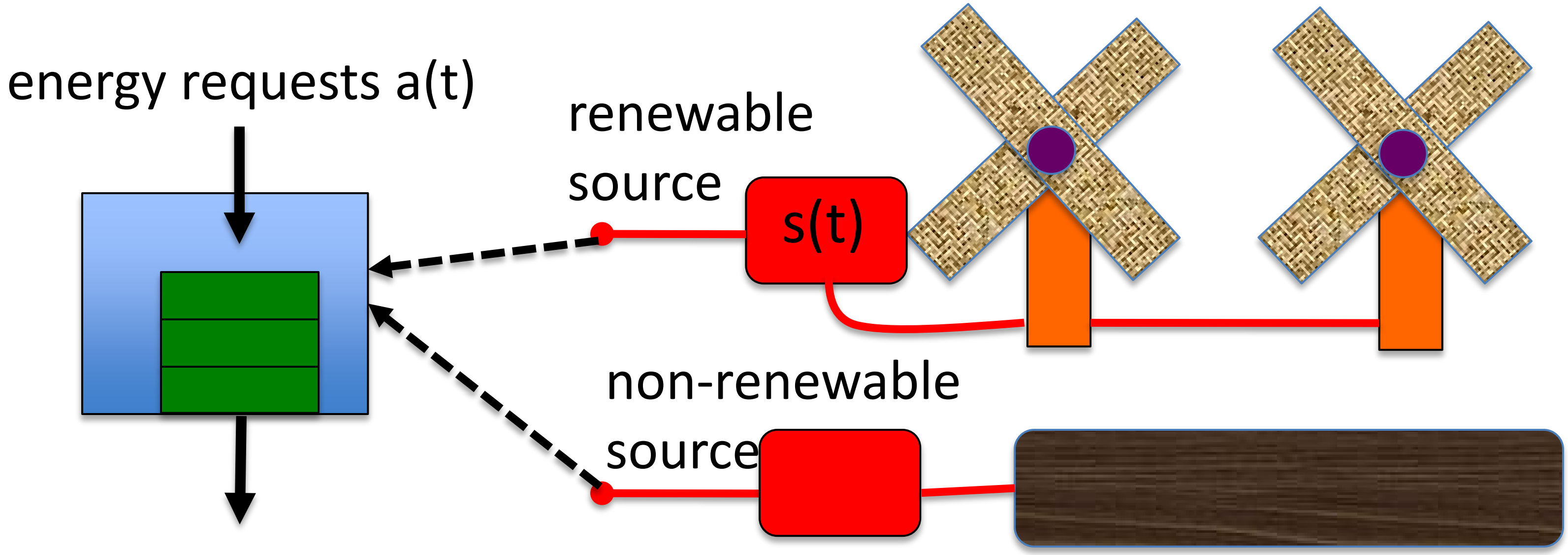
Efficient Algorithms for Renewable Energy Allocation to Delay Tolerant Consumers



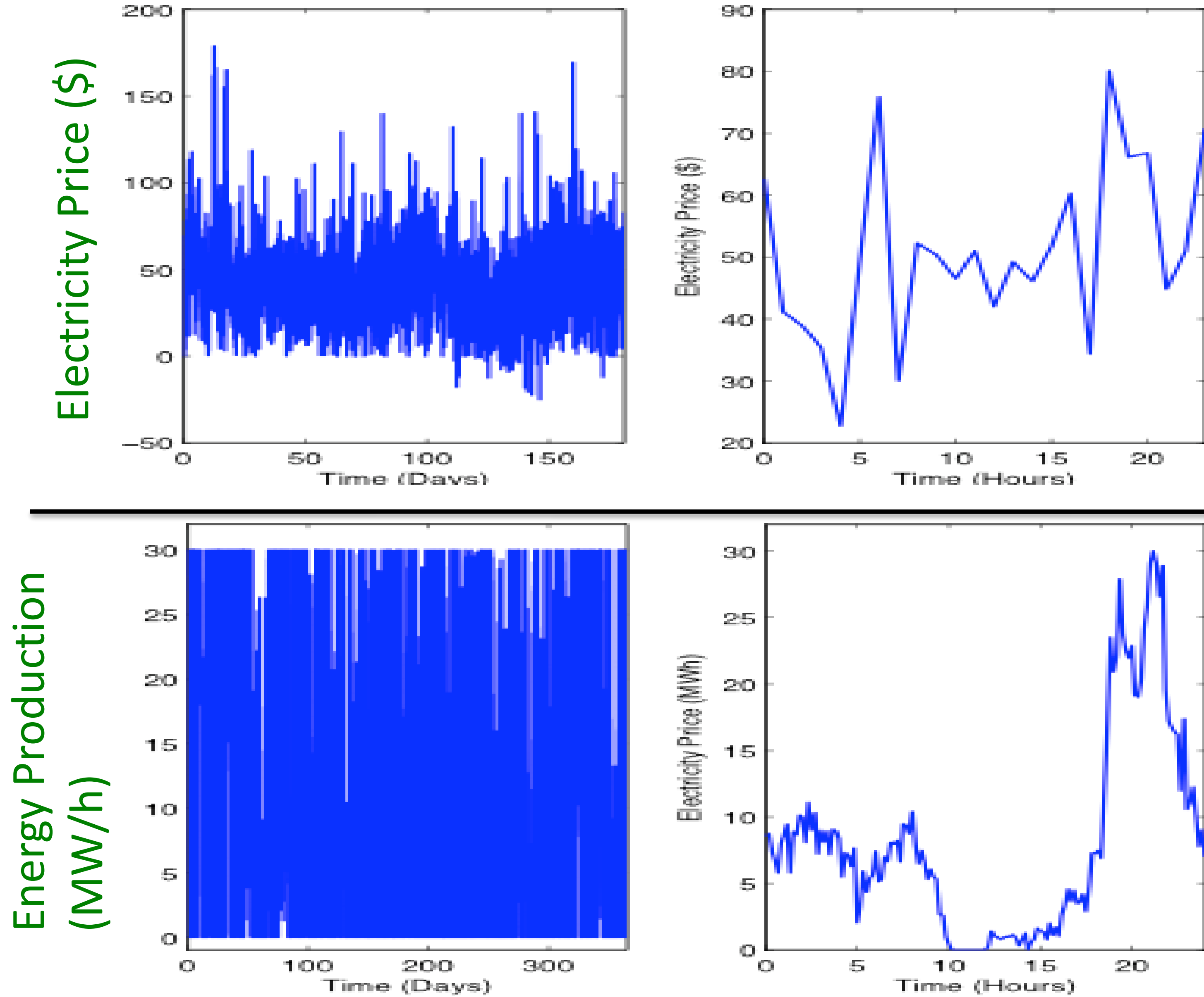
Michael J. Neely , Arash Saber Tehrani , Alexandros G. Dimakis
University of Southern California

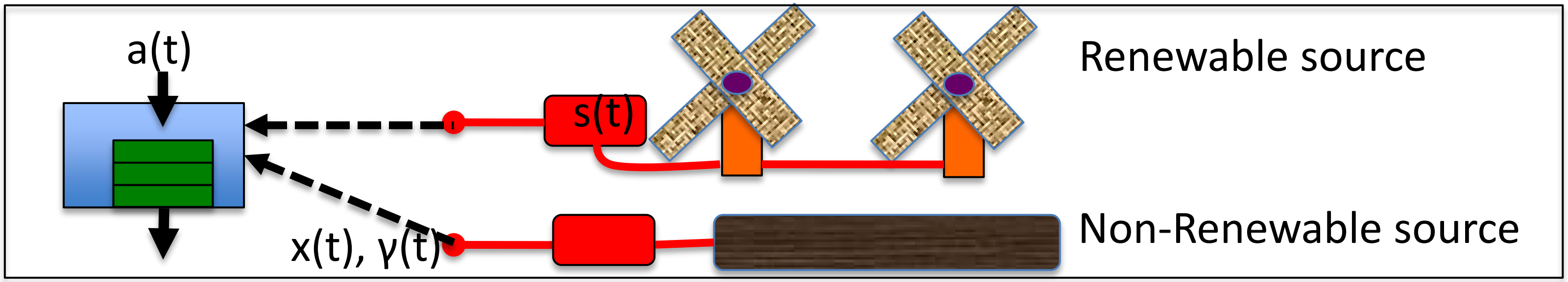
*Paper to appear at: 1st IEEE International Conf. on Smart Grid Communications, 2010
PDF on Stochastic Network Optimization Homepage: http://ee.usc.edu/stochastic-nets/
*Sponsored in part by the NSF Career CCF-0747525



- Renewable sources of energy can have ***variable*** and ***unpredictable*** supplies s(t).
- We can integrate renewable sources more easily if consumers tolerate service within some ***maximum allowable delay*** D_{max}.
- Might sometimes need to purchase energy from non-renewable source to meet the deadlines, and ***purchase price can be highly variable***.

Example Data: (Top Row) Spot Market Price
(Bottom Row) Energy Production in a California Wind Turbine





Talk Outline:

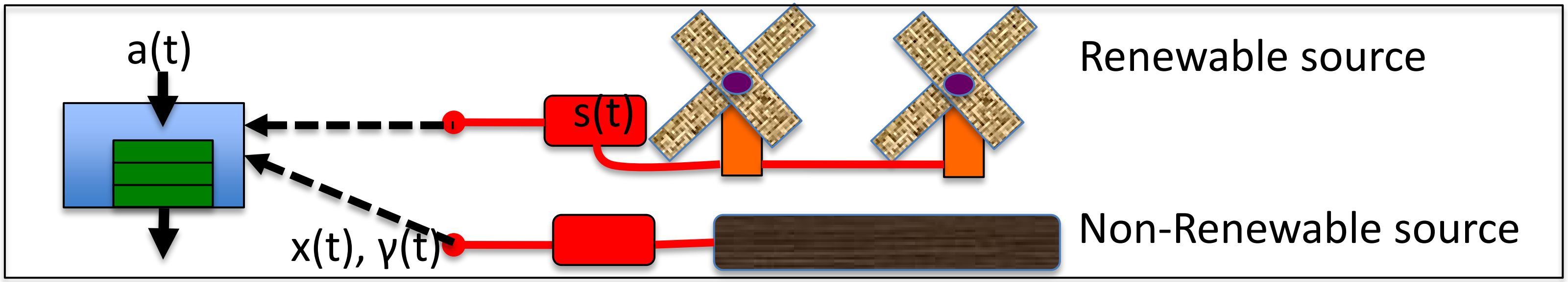
- First Problem: Minimize time average cost of purchasing non-renewable energy (i.i.d. case)

- Second Problem: Joint pricing of customers and purchasing of non-renewables (i.i.d. case).

- Generalize to ***arbitrary sample paths*** using "Universal Scheduling Theory" of Lyapunov Optimization.

- Simulation results using CAISO spot market prices $\gamma(t)$ and 10-minute energy production $s(t)$ from Western Wind resources Dataset (from National Renewable Energy Lab).

Problem 1: Minimize Average Cost of Non-Renewable Purchases



- Slotted Time: t = {0, 1, 2, …}
- a(t) = energy requests on slot t (serve with max delay D_{max}).
- s(t) = renewable energy supply on slot t. ("use-it-or-loose-it")
- x(t) = amount non-renewable energy purchased on slot t.
- γ(t) = $$/unit energy price of non-renewables on slot t.
- Q(t) = Energy request queue

requests (random)

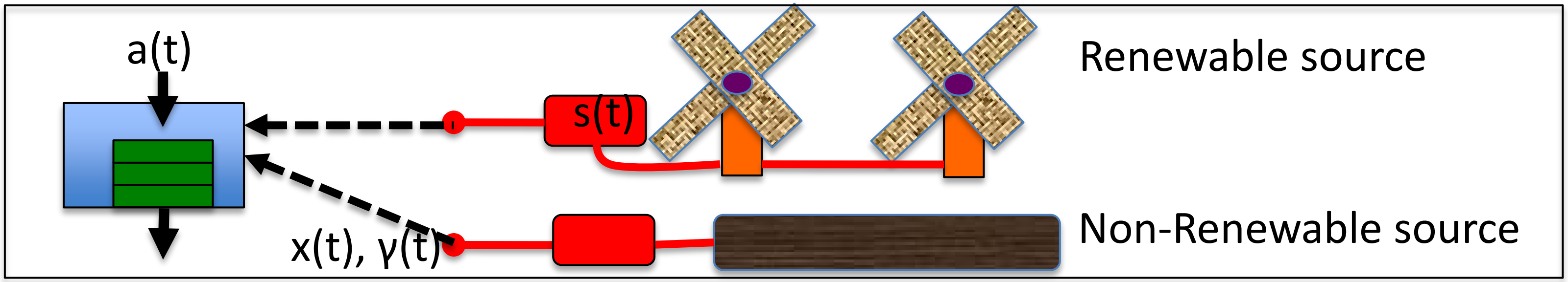
$$\boxed{Q(t+1) = \max[Q(t) - s(t) - x(t), 0] + a(t) \quad , \qquad cost(t) = x(t)\gamma(t)}$$

Renewable supply (random) (use-it-or-loose-it)

Non-Renewables purchased (decision variable)

purchase price (random)

Problem 1: Minimize Average Cost of Non-Renewable Purchases



$$Q(t+1) = \max[Q(t) - s(t) - x(t), 0] + a(t) \quad , \qquad cost(t) = x(t)\gamma(t)$$

Assumptions:

- For all slots t we have:

  $0 \leq a(t) \leq a_{max}$, $0 \leq s(t) \leq s_{max}$, $0 \leq \gamma(t) \leq \gamma_{max}$, $0 \leq x(t) \leq x_{max}$

- x_{max} units of energy always available for purchase from non-renewable (but at variable price $\gamma(t)$).

- $a_{max} \leq x_{max}$ (possible to meet all demands in 1 slot at high cost)

- $(a(t), s(t), \gamma(t))$ vector is i.i.d. over slots with ***unknown distribution***

Problem 1: Minimize Average Cost of Non-Renewable Purchases

$$\boxed{Q(t+1) = \max[Q(t) - s(t) - x(t), 0] + a(t) \quad , \qquad cost(t) = x(t)\gamma(t)}$$

Possible formulation via Dynamic Programming (DP):

"Minimize average cost subject to max-delay D_{max}."

- This can be written as a DP, but requires distribution knowledge.
- Recent work on delay tolerant electricity consumers using DP is: [Papavasiliou and Oren, 2010]

We will not use DP. We will take a different approach...

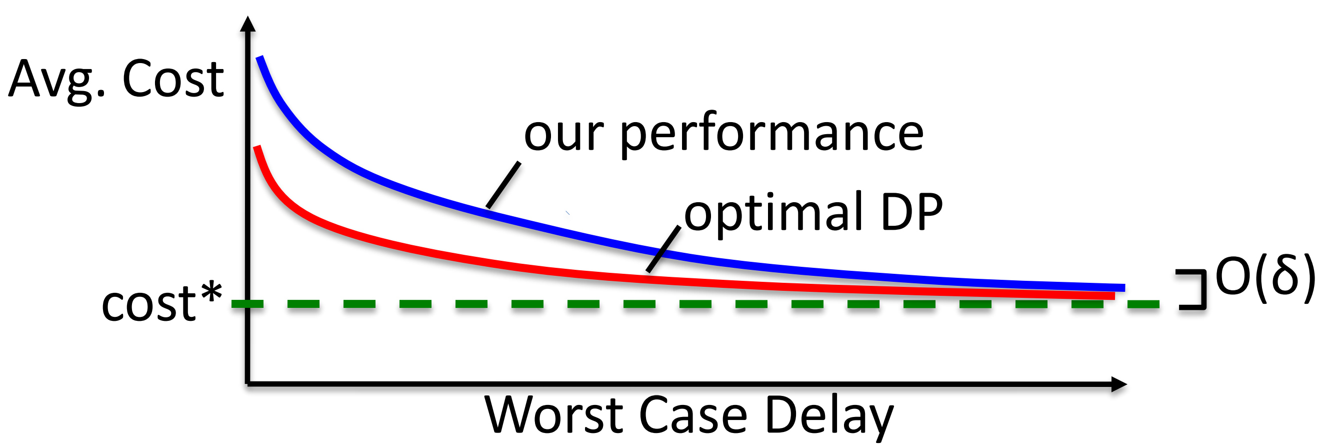
Problem 1: Minimize Average Cost of Non-Renewable Purchases

$$Q(t+1) = \max[Q(t) - s(t) - x(t), 0] + a(t) \quad , \qquad cost(t) = x(t)\gamma(t)$$

Relaxed Formulation via Lyapunov Optimization for Queue Networks:

Minimize: E{cost} (time average)
Subject to: (1) E{Q} < infinity (a "queue stability" constraint)
(2) $0 \leq x(t) \leq x_{max}$ for all t

- Define cost* = min cost subject to stability
- By definition: cost* ≤ cost delivered by ***any other alg*** (including DP)
- We will get within O(δ) of cost*, with ***worst-case delay*** of 1/δ.



Advantages of Lyapunov Optimization for Queueing Networks:

- No knowledge of distribution information is required.
- Explicit [O(δ), O(1/δ)] performance guarantees.
- Robust to changes in statistics, arbitrary correlations, non-ergodic, arbitrary sample paths (***as we will show in this work***).
- Worst case delay bounds (***as we will show in this work***).
- No curse of dimensionality: Implementation is just as easy in extended formulations with many dimensions:

General Lyapunov Optimization Problem: [Georgiadis, Neely, Tassiulas, F&T 2006]

Minimize : $E\{y\}$

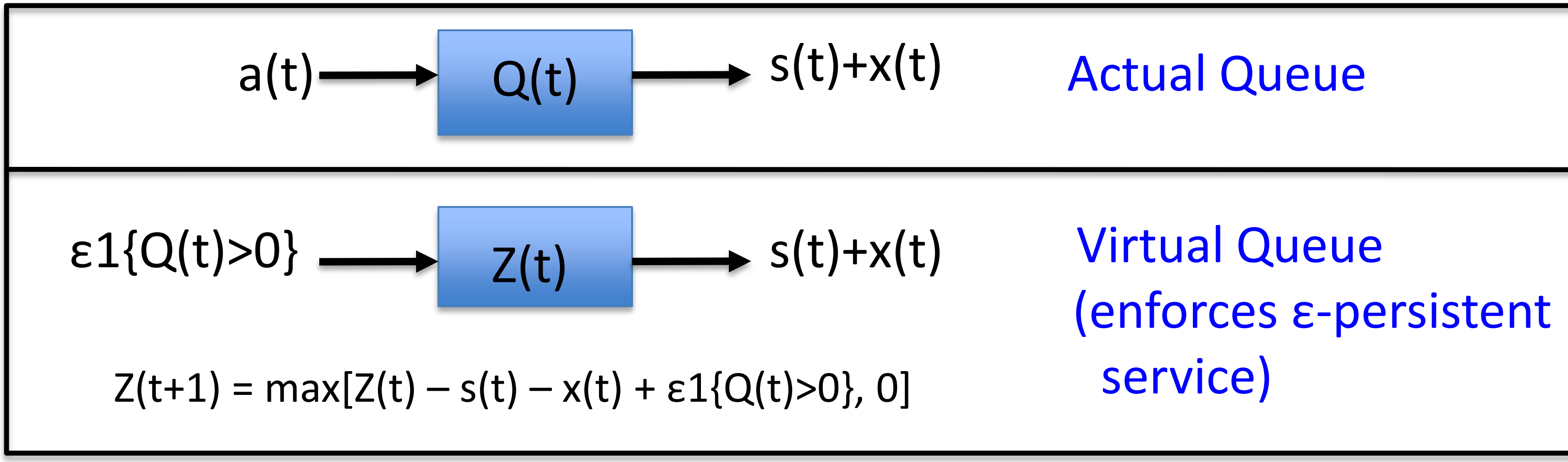
Subject to: (1) $E\{x_i\} \leq 0$ for all i in {1, …, N}

(2) Queue k is stable for all k in {1, …, K}

(3) Control action on slot t in *ActionSpace*(t)

(for all t in {0, 1, 2, …})

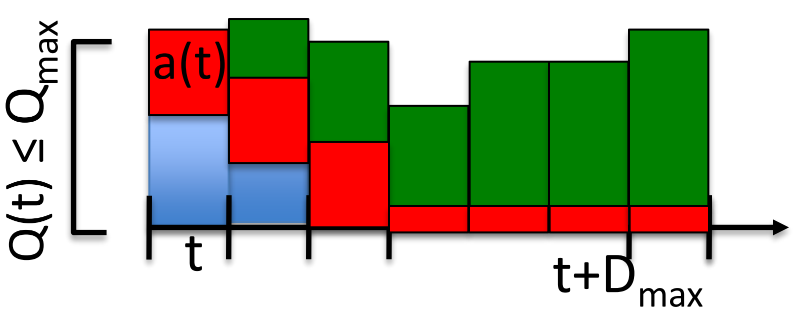
Virtual Queue for Worst-Case Delay Guarantee (fix ε>0):



Theorem: Any algorithm with bounded queues $Q(t) \leq Q_{max}$, $Z(t) \leq Z_{max}$ for all t yields worst-case delay of:

$$D_{max} = \left\lceil \frac{Q_{max} + Z_{max}}{\varepsilon} \right\rceil \text{ slots}$$

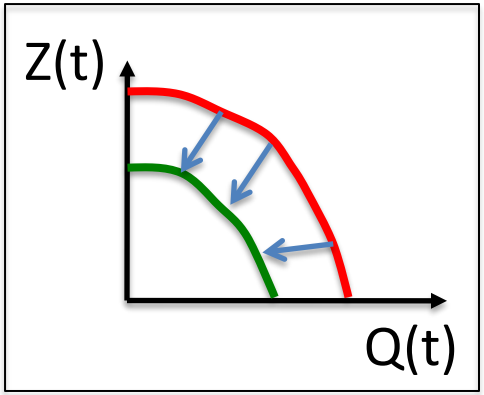
Proof Sketch: Suppose not. Consider slot t, a(t):



Then: $$\sum_{\tau=t}^{t+D_{max}} [s(\tau)+x(\tau)] \leq Q_{max}$$

Implies: $Z(t+D_{max}) > Z_{max}$ (contradiction)

Stabilize Z(t) and Q(t) while minimizing average cost cost(t):



Lyapunov Function: $L(t) = Z(t)^2 + Q(t)^2$

Lyapunov Drift: $\Delta(t) = E\{L(t+1) - L(t) | Z(t), Q(t)\}$

Take actions to greedily minimize "***Drift-Plus-Weighted-Penalty***":

$$\boxed{\text{Minimize: } \Delta(t) + V\gamma(t)x(t)}$$

where V is a postiive constant that affects the [O(1/V), O(V)] Cost-delay tradeoff.

(using V=1/δ recovers the [O(δ), O(1/δ)] tradeoff.)

Resulting Algorithm: Every slot t, observe $(Z(t), Q(t), \gamma(t))$. Then:

- Choose $x(t) = \begin{cases} 0, & \text{if } Q(t) + Z(t) \leq V\gamma(t) \\ x_{max}, & \text{if } Q(t) + Z(t) > V\gamma(t) \end{cases}$

- Update virtual queues $Q(t)$ and $Z(t)$ according to their equations

Define: $Q_{max} = V\gamma_{max} + a_{max}$, $Z_{max} = V\gamma_{max} + \varepsilon$

Theorem: Under the above algorithm:

(a) $Q(t) \leq Q_{max}$, $Z(t) \leq Z_{max}$ for all t.

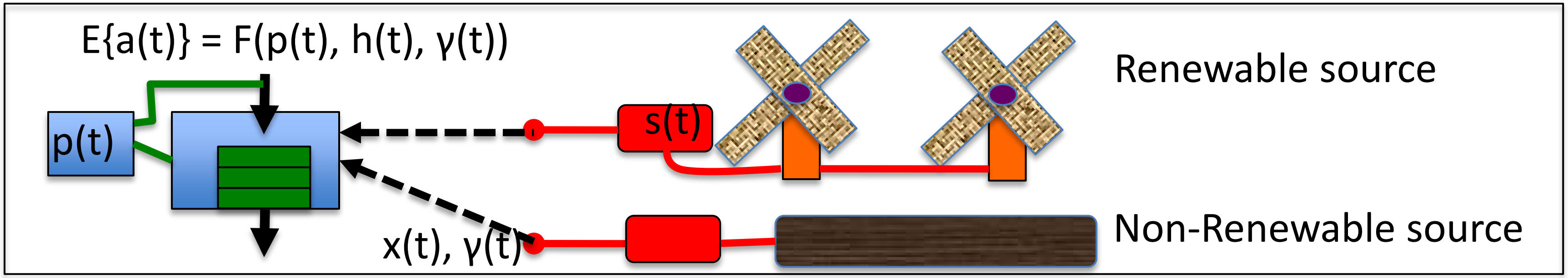
(b) Delay $\leq (Q_{max} + Z_{max})/\varepsilon = O(V)$

Further, if $(s(t), a(t), \gamma(t))$ i.i.d. over slots, and if $\varepsilon \leq \max[E\{a(t)\}, E\{s(t)\}]$
Then:

$$E\{cost\} \leq cost^* + B/V$$

[where $B = (s_{max} + x_{max})^2 + a_{max}^2 + \varepsilon^2$]

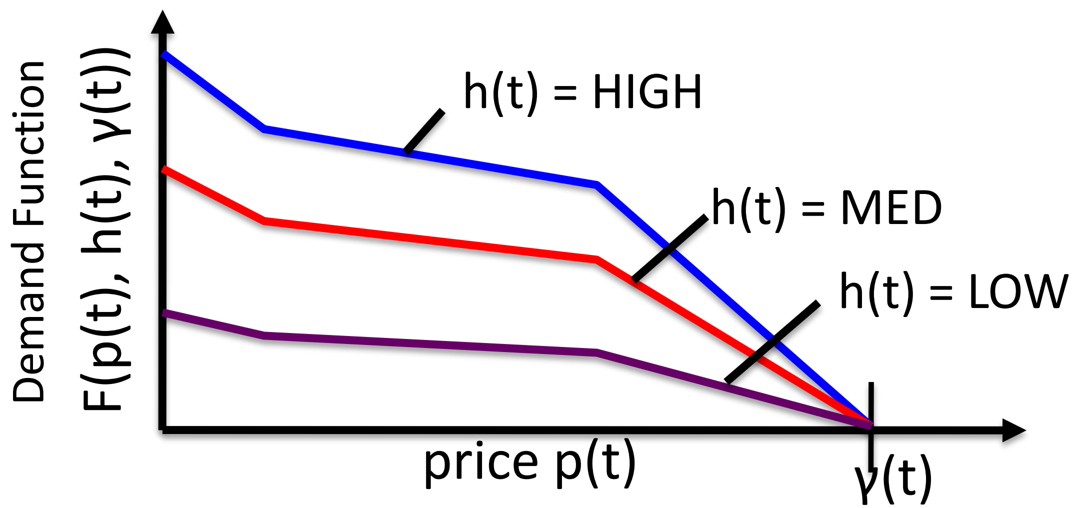
Problem 2: Joint Pricing and Energy Allocation



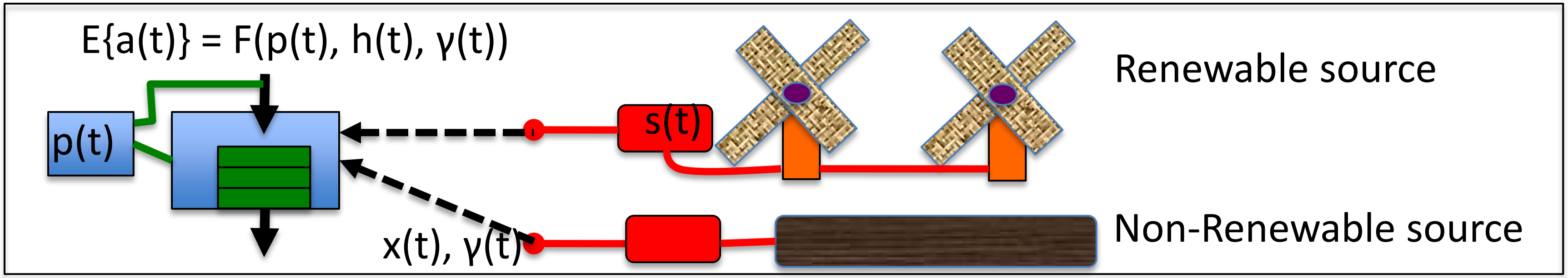
Same system model, with following extensions:

- a(t) = arrivals = Random function of pricing decision p(t)
- h(t) = additional "demand state" (e.g. "HIGH, MED, LOW")
- E{a(t) | p(t), h(t), γ(t)} = F(p(t), h(t), γ(t)) = Demand Function

Example:



Problem 2: Joint Pricing and Energy Allocation



Same system model, with following extensions:

- $a(t)$ = arrivals = Random function of pricing decision $p(t)$
- $h(t)$ = additional "demand state" (e.g. "HIGH, MED, LOW")
- $E\{a(t)\,|\,p(t), h(t), \gamma(t)\} = F(p(t), h(t), \gamma(t))$ = Demand Function

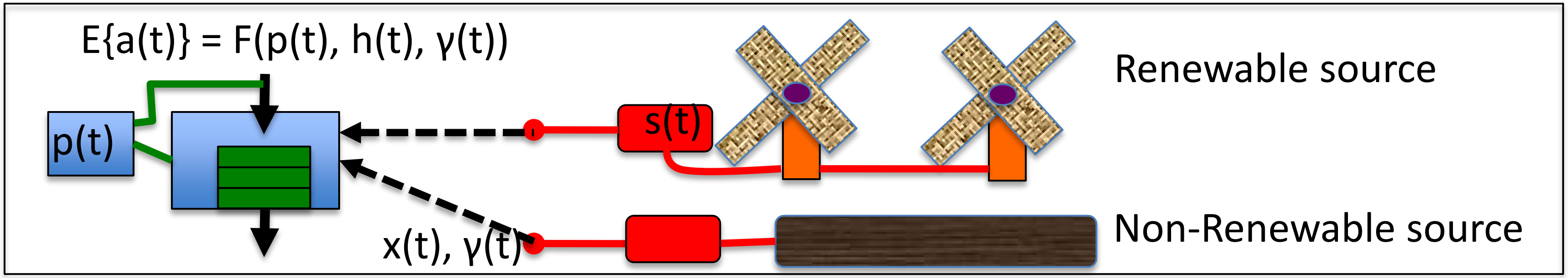
New Problem:

- Profit(t) = $a(t)p(t) - x(t)\gamma(t)$

- Maximize Time Average Profit!

- Profit* = Optimal Time Avg. Profit Subject to Stability

Problem 2: Joint Pricing and Energy Allocation



Drift-Plus-Penalty for New Problem:

$$\Delta(t) - VE\{Profit(t)|Z(t),Q(t)\} = \Delta(t) - VE\{a(t)p(t) - x(t)\gamma(t)|Z(t),Q(t)\}$$

Resulting Algorithm:

Every slot t, observe $(h(t), Z(t), Q(t), \gamma(t))$. Then:

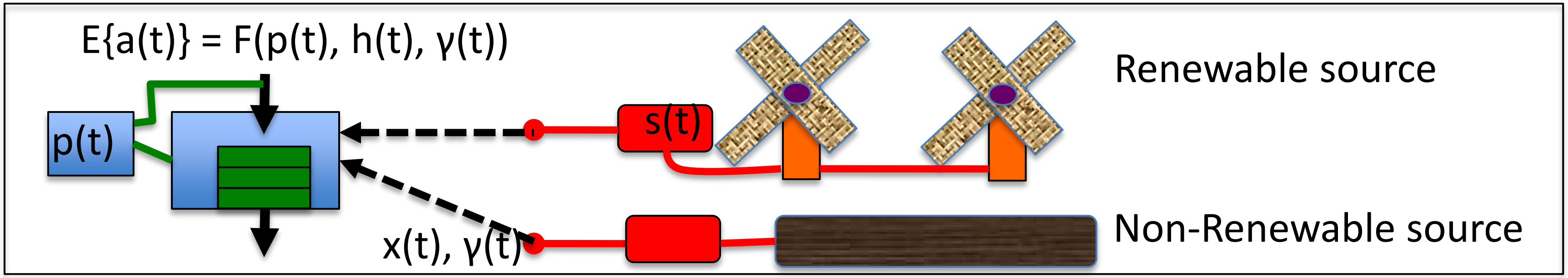
•(Pricing) Choose $p(t)$ in $[0, p_{max}]$ to solve:

Maximize:	$F(p(t),h(t),\gamma(t))(Vp(t) - Q(t))$
Subject to:	$0 \leq p(t) \leq p_{max}$

•(Purchasing) Choose $x(t)$ same as before.

•Update queues $Q(t)$, $Z(t)$ same as before.

Problem 2: Joint Pricing and Energy Allocation



Drift-Plus-Penalty for New Problem:

$$\Delta(t) - VE\{Profit(t)|Z(t),Q(t)\} = \Delta(t) - VE\{a(t)p(t) - x(t)\gamma(t)|Z(t),Q(t)\}$$

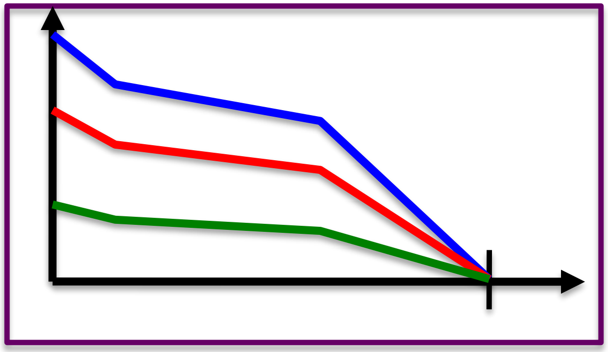
Resulting Algorithm:

Every slot t, observe $(h(t), Z(t), Q(t),\gamma(t))$. Then:

- (Pricing) Choose $p(t)$ in $[0, p_{max}]$ to solve:

Maximize: $F(p(t),h(t),\gamma(t))(Vp(t) - Q(t))$

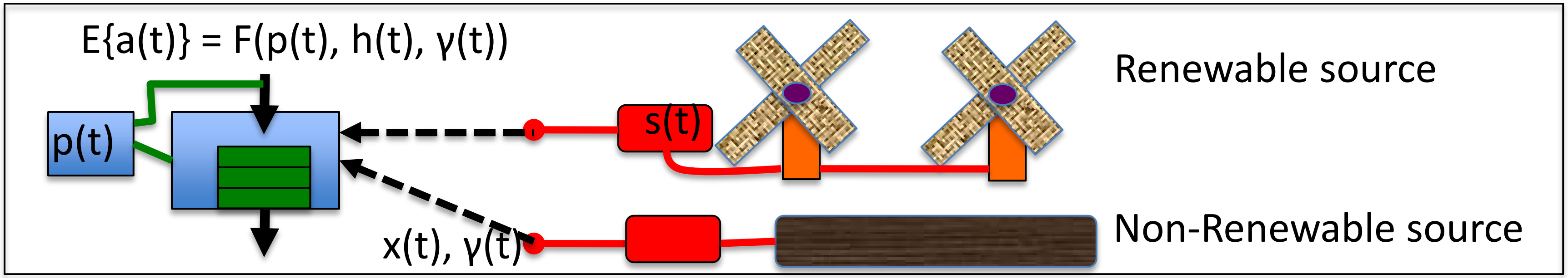
Subject to: $0 \leq p(t) \leq p_{max}$



- (Purchasing) Choose $x(t)$ same as before.
- Update queues $Q(t)$, $Z(t)$ same as before.

*If $F(p,h,\gamma) = \beta(h)G(p,\gamma)$, don't need to know demand state $h(t)$!

Problem 2: Joint Pricing and Energy Allocation



Drift-Plus-Penalty for New Problem:

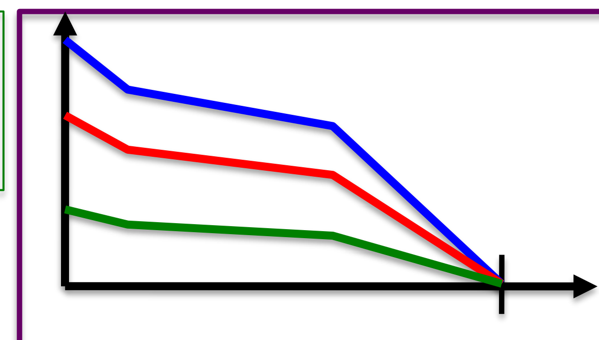
$$\Delta(t) - VE\{\text{Profit}(t)\,|\,Z(t),Q(t)\} = \Delta(t) - VE\{a(t)p(t) - x(t)\gamma(t)\,|\,Z(t),Q(t)\}$$

Resulting Algorithm:

Every slot t, observe $(h(t), Z(t), Q(t), \gamma(t))$. Then:

- (Pricing) Choose $p(t)$ in $[0, p_{max}]$ to solve:

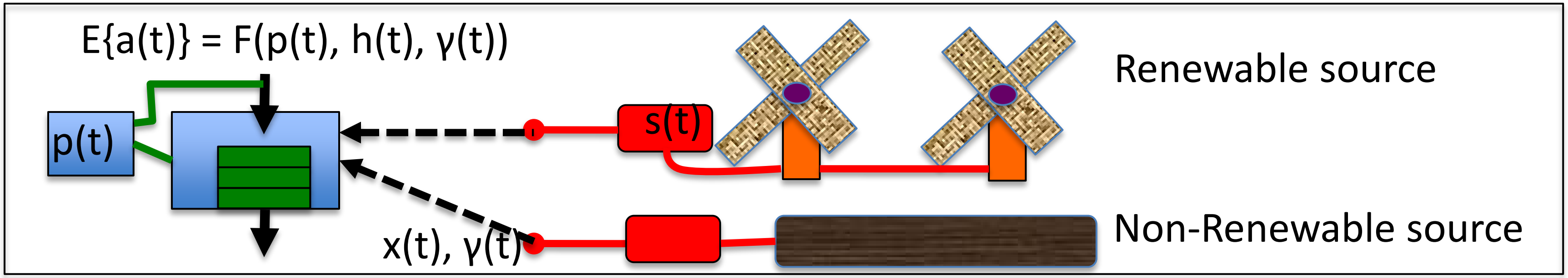
$$\text{Maximize: } \beta(h(t))G(p(t),\gamma(t))(Vp(t) - Q(t))$$
$$\text{Subject to: } 0 \leq p(t) \leq p_{max}$$



- (Purchasing) Choose $x(t)$ same as before.
- Update queues $Q(t)$, $Z(t)$ same as before.

*If $F(p,h,\gamma) = \beta(h)G(p,\gamma)$, don't need to know demand state $h(t)$!

Problem 2: Joint Pricing and Energy Allocation



Theorem: Under the joint pricing and energy allocation algorithm:

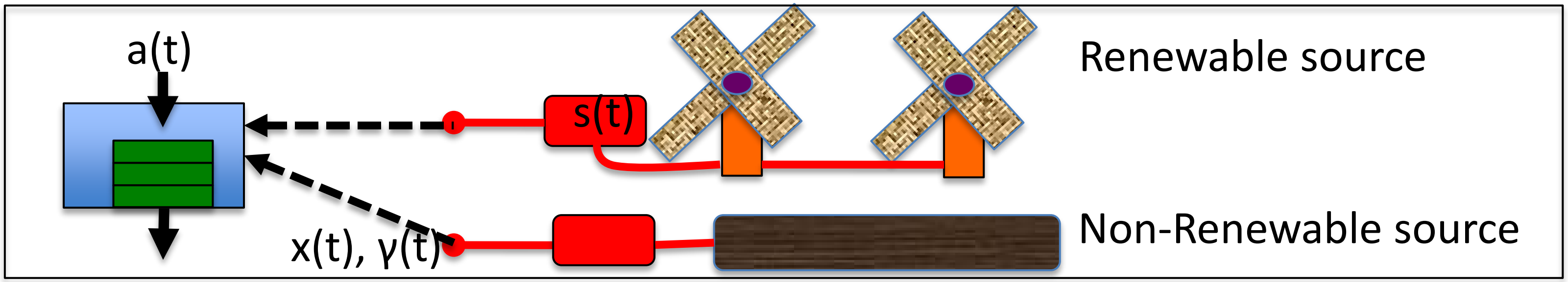
(a) Worst case queue bounds Q_{max}, Z_{max} same as before.

(b) Worst case delay bound D_{max} same as before, i.e., $O(V)$.

(c) If $(s(t), \gamma(t), h(t))$ i.i.d. over slots, and $\varepsilon \leq E\{s(t)\}$, then:

$$E\{profit\} \geq profit^* - O(1/V)$$

Universal Scheduling for Arbitrary Sample Paths…



Consider the first problem again (x(t) = only decision variable):

Suppose (s(t), γ(t), a(t)) have ***arbitrary sample path!***

(assume they are still bounded: $[0, s_{max}]$, $[0, \gamma_{max}]$, $[0, a_{max}]$.)

Universal Scheduling Theorem:

(a) Worst case queue bounds Q_{max}, Z_{max} same as before.

(b) Worst case delay bound D_{max} same as before, i.e., O(V).

(c) For any integers T>0, R>0:

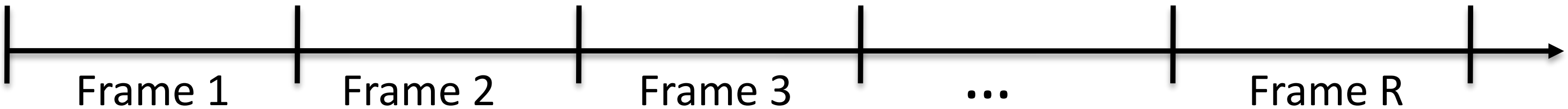
$$\boxed{\frac{1}{RT}\sum_{t=0}^{RT-1} x(t)\gamma(t) \leq \frac{1}{R}\sum_{r=0}^{R-1} C_r^* + BT/V}$$

"Genie-Aided" T-Slot Lookahead Cost!

For every R>0, T>0:

$$\frac{1}{RT}\sum_{t=0}^{RT-1} x(t)\gamma(t) \leq \frac{1}{R}\sum_{r=0}^{R-1} C_r^* + BT/V$$

R frames of size T slots:



T-Slot Lookahead Problem for frame r in {0, …, R-1}:

c_r^* computed below, ***assuming future values of (a(τ), s(τ), γ(τ)) are fully known*** in frame r:

$$\text{Minimize:} \quad c_r^* \triangleq \frac{1}{T}\sum_{\tau=rT}^{(r+1)T-1} \gamma(\tau)x(\tau)$$

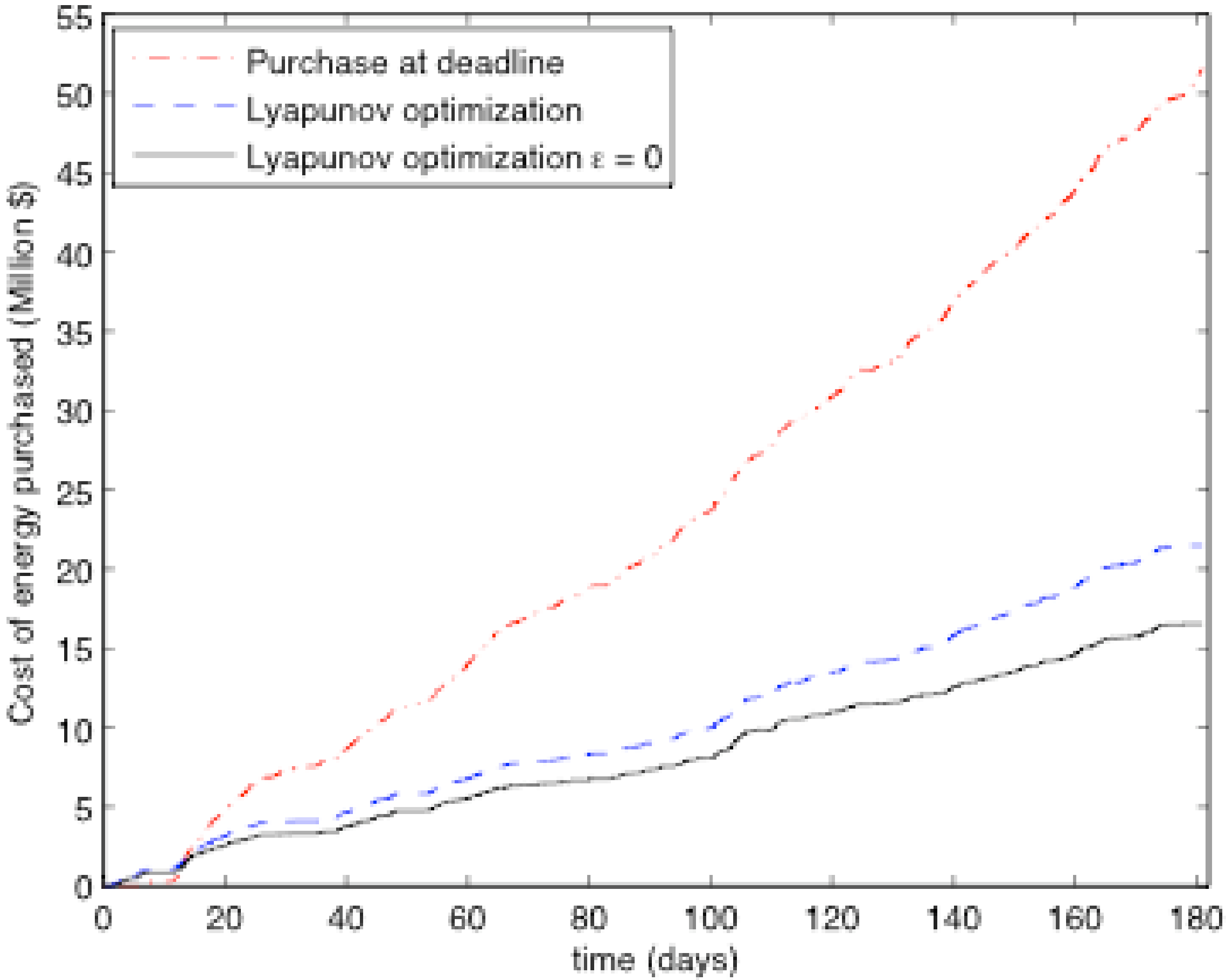
$$\text{Subject to:} \quad (1) \quad \sum_{\tau=rT}^{(r+1)T-1} [s(\tau)+x(\tau)-a(\tau)] \geq 0$$

$$(2) \quad \sum_{\tau=rT}^{(r+1)T-1} [s(\tau)+x(\tau)-\epsilon] \geq 0$$

$$(3) \quad 0 \leq x(\tau) \leq x_{max} \forall \tau \in \{rT, \ldots, (r+1)T-1\}$$

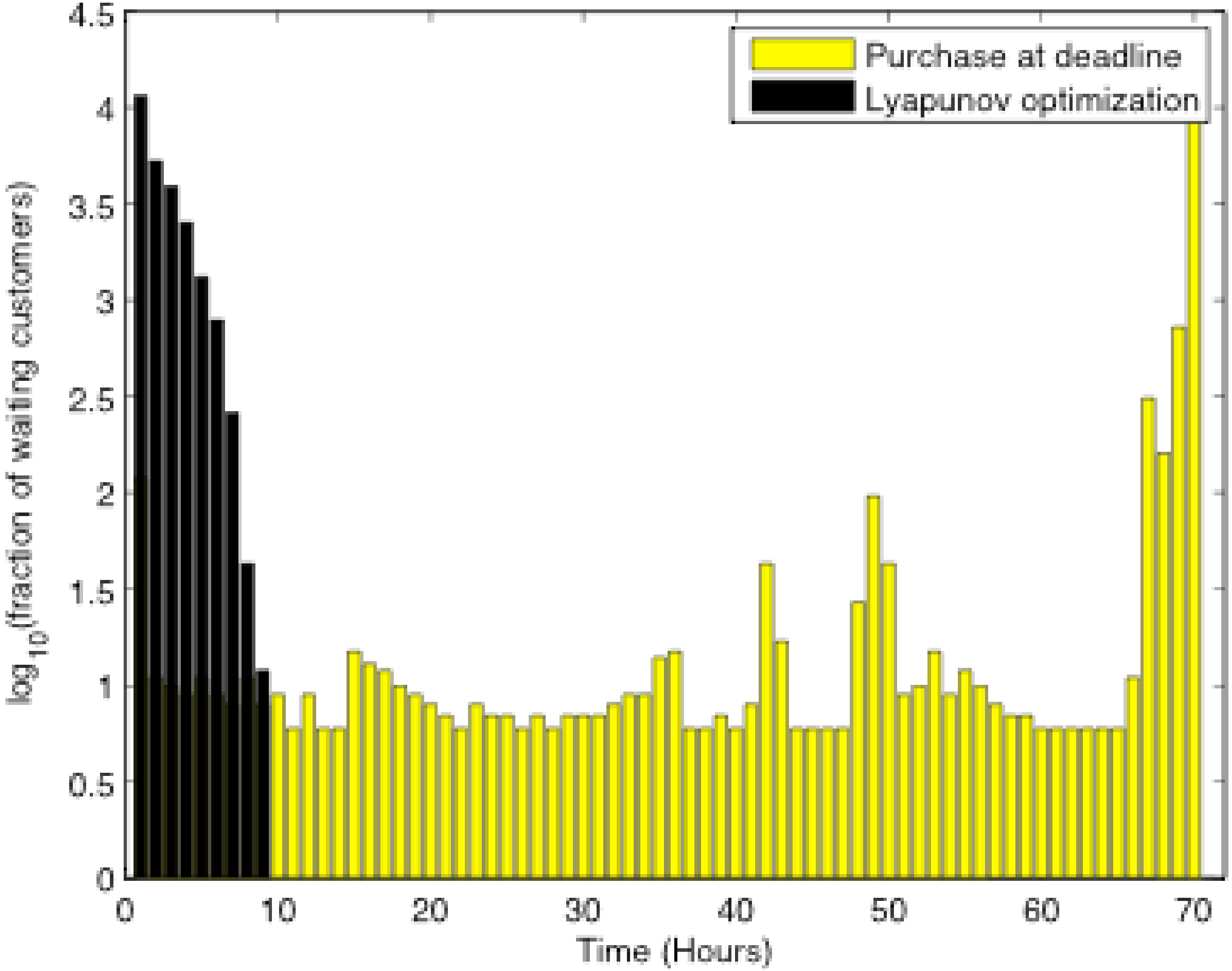
Simulations over Real Data Sets:

- We used 10 minute slot sizes (granularity of the available data)
- Compare to simple “Purchase at Deadline” algorithm.
- We chose V=100 ➔ D_{max} = 400 slots (70 hours)

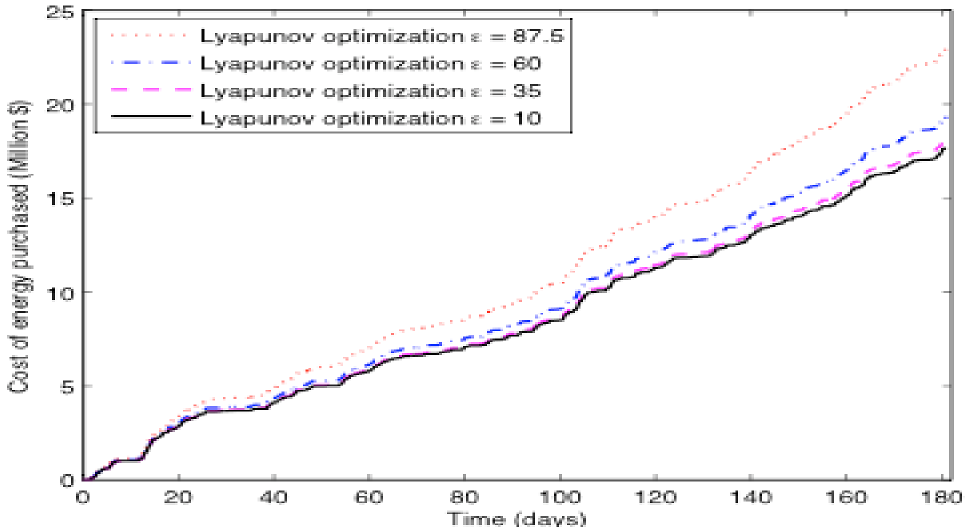


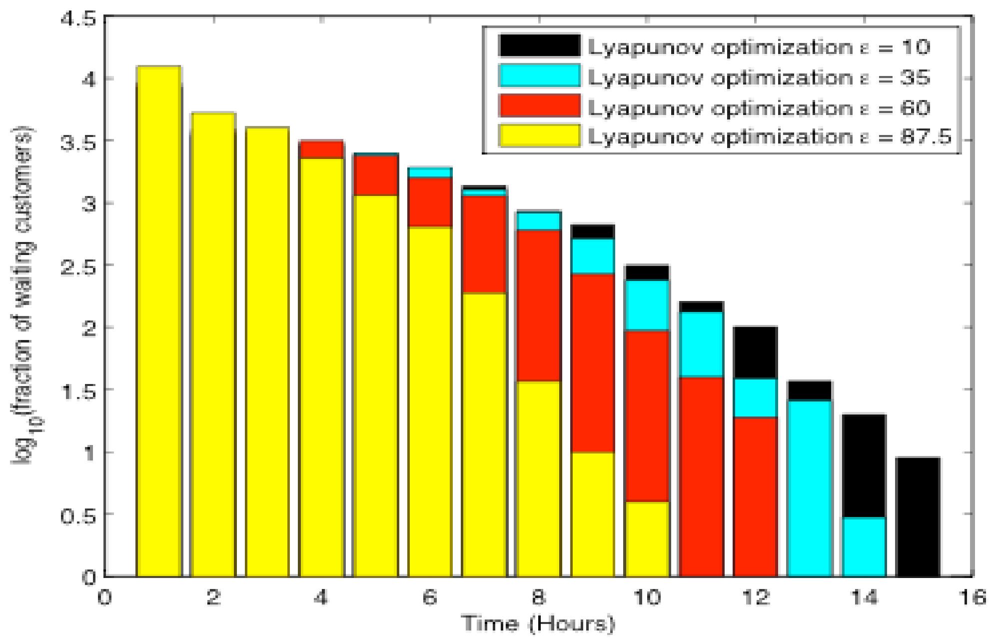
Same experiment: Histogram of Delay (V=100, ε= 87.5):

Our algorithm yields worst-case delay considerably less than the bound D_{max}. Worst case observed delay was 60 slots (10 hours)

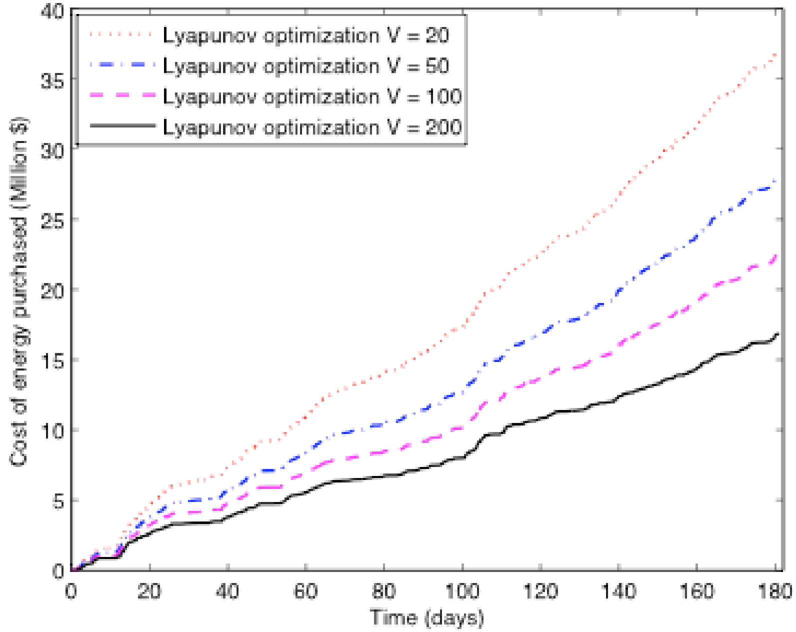


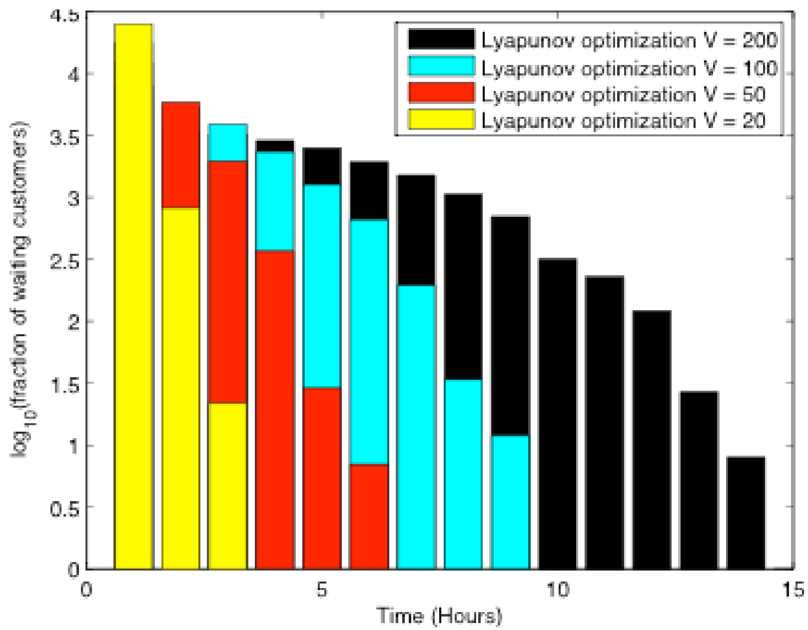
Some more simulations: Changing the ε parameter:



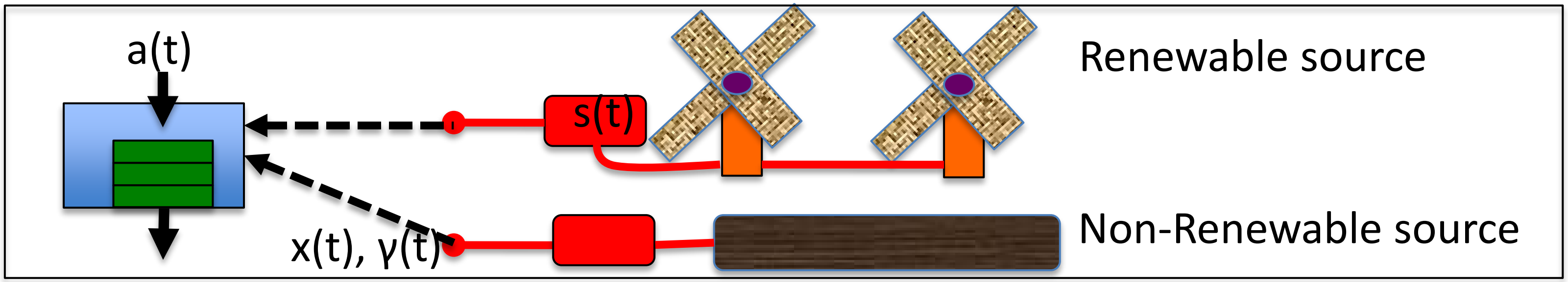


Some more simulations: Changing the V parameter:

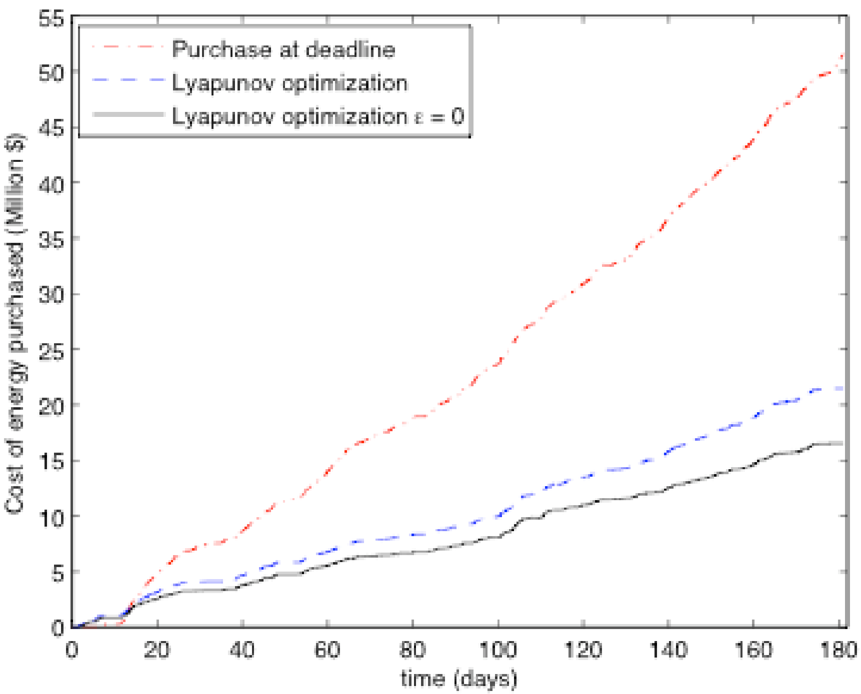


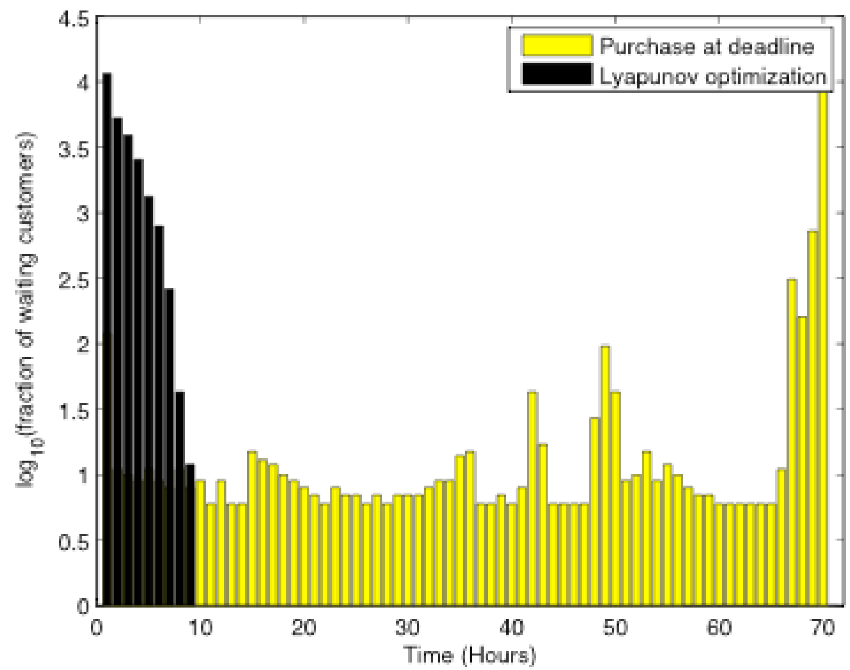


Concluding Slide:



- Lyapunov Optimization for Renewable Energy Allocation
- No need to know distribution. Robust to arbitrary sample paths.
- Explicit [O(1/V), O(V)] performance-delay tradeoff





Explanation of Why Delay is small even with $\varepsilon=0$...

Even with $\varepsilon=0$, we still get the same Q_{max} bound. ($Q(t) \leq Q_{max}$ for all t).

Delay of requests that arrive on slot t is equal to the smallest integer T such that:

$$\sum_{\tau=t}^{t+T} [s(\tau)+x(\tau)] \geq Q(t)$$

So delay will be less than or equal to T whenever:

$$\sum_{\tau=t}^{t+T} s(\tau) \geq Q_{max}$$

There is no guarantee on how long this will take for arbitrary s(t) processes, but one can compute probabilities of exceeding a certain value if we try to use a stochastic model for s(t).