# Efficient Spectrum Allocation and Auction for Wireless Networks

**XiangYang Li**[1]
xli@cs.iit.edu

[1]Department of Computer Science
Illinois Institute of Technology

April 6, 2009

# Organization of Talk I

# Organization of Talk II

- Non-instantaneous requests with $\beta = 1$
- Non-instantaneous requests with $\beta > 1$

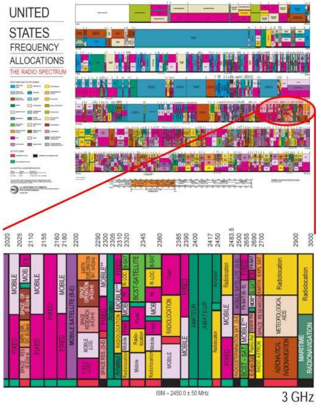
- Upperbounds
  - $\beta = 1$
  - $\beta > 1$
- Experiments

5. Conclusions

The results presented here are collaborated with my student Ping Xu at IIT.

# Widespread of Wireless Devices: Need Spectrum

# Traditionally: Fixed Spectrum Allocation




- Fixed spectrum allocation traditionally;
- ISM band: industrial, scientific and medical (ISM) radio bands.
  - WLAN: Bluetooth, 802.11
  - Cordless phones
  - RFID

# Pros and Cons of Fixed Allocation

- Pros: easy to manage
- Cons: White Space (spectral, temporal, and spatial)
  Even in more congested areas, there is still ample space [1]
  - Dallas – **40** percent
  - Boston – **38** percent
  - Seattle – **52** percent
  - San Francisco – **37** percent.

[1] from www.tvtechnology.com

# Pros and Cons of Fixed Allocation

- Pros: easy to manage
- Cons: White Space (spectral, temporal, and spatial)
  Even in more congested areas, there is still ample space [1]
  - Dallas – **40** percent
  - Boston – **38** percent
  - Seattle – **52** percent
  - San Francisco – **37** percent.

[1] from www.tvtechnology.com

# More Spectrum By Technology Changes

The planned switchover to digital television may free up large areas between **54MHz** and **698MHz**.

- "Battle Heats Up for TV Spectrum White Space Use" – WIMAX.com

On November 4, 2008 the FCC: unlicensed and free use of TV white space frequencies for all.

Exact amount depends on

1. Broadcast TV channels going on and off the air
2. Wireless microphone users registering for protected status
3. Changes in White Space rules and regulations

# New Spectrum Usage Technologies

To improve spectrum usage:

- Dynamic Spectrum Allocation: allocated when needed
- Opportunistic Spectrum Usage: use it when no interference
  - Software defined radio,
  - Cognitive Radio (Licensed Band Cognitive Radio and Unlicensed Band Cognitive Radio)

# New Spectrum Usage Technologies

To improve spectrum usage:

- Dynamic Spectrum Allocation: allocated when needed
- Opportunistic Spectrum Usage: use it when no interference
  - Software defined radio,
  - Cognitive Radio (Licensed Band Cognitive Radio and Unlicensed Band Cognitive Radio)

# New Spectrum Usage Technologies

To improve spectrum usage:

- Dynamic Spectrum Allocation: allocated when needed
- Opportunistic Spectrum Usage: use it when no interference
  - Software defined radio,
  - Cognitive Radio (Licensed Band Cognitive Radio and Unlicensed Band Cognitive Radio)

Challenges of opportunistic spectrum usage:
How to deal with selfish behavior?
Need combine game theory with wireless communication modeling

# New Spectrum Usage Technologies

To improve spectrum usage:

- Dynamic Spectrum Allocation: allocated when needed
- Opportunistic Spectrum Usage: use it when no interference
  - Software defined radio,
  - Cognitive Radio (Licensed Band Cognitive Radio and Unlicensed Band Cognitive Radio)
- Market Driven Approach: short-term lease, users bid for spectrum usage.

# New Spectrum Usage Technologies

To improve spectrum usage:

- Dynamic Spectrum Allocation: allocated when needed
- Opportunistic Spectrum Usage: use it when no interference
  - Software defined radio,
  - Cognitive Radio (Licensed Band Cognitive Radio and Unlicensed Band Cognitive Radio)
- Market Driven Approach: short-term lease, users bid for spectrum usage.

# Market Driven Approach

Construct an allocation/auction to assign spectrum

- Auctioneer: central authority represents primary users
- Bidders: secondary users, selfish but rational

We need determine winners and payments with objectives

- Maximize the social efficiency - total valuation of winners , or
- Maximize the revenue - total payment collected from bidders.

# Market Driven Approach

Construct an allocation/auction to assign spectrum

- Auctioneer: central authority represents primary users
- Bidders: secondary users, selfish but rational

We need determine winners and payments with objectives

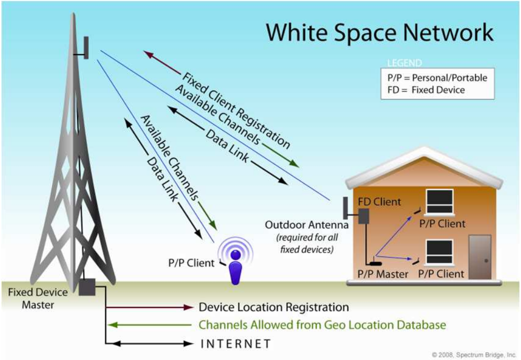
- Maximize the social efficiency - total valuation of winners, or
- Maximize the revenue - total payment collected from bidders.

Models, models, and models, ☺

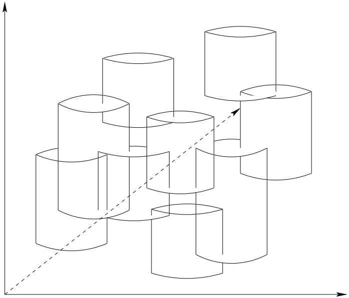
What is our network model, the bidding model and more....

# Bidding and Network Model

Primary user $\mathcal{U}$ who holds the right of some spectrum channels



# Time-Space model for Single Channel



- secondary users $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ who wants to lease the right of some spectrum channels
  - arrived at time $a_i$
  - in some geometry region $D(v_i, r_i)$
  - for some time period $T_i$
  - for some frequencies $F_i$
  - with bid $b_i$,

In summary, a bidding by a user $v_i$ can be written as follows

$$B_i = [b_i, a_i, F_i, D(v_i, r_i), T_i]$$

# Objective of Spectrum Allocation/Auction

Find an allocation of spectrum to users, which must be

- conflict free in geometry region, time period and frequencies
- maximize the social efficiency $\sum_{i=1}^{m} x_i b_i$, where $b_i$ is the true valuation if the mechanism is truthful.

This problem is obviously **NP-hard**.

# When should we make decisions?

The decisions could be

- offline: make decisions after know all requests;
- online: make decisions when requests arrived.

# When should we make decisions?

The decisions could be

- offline: make decisions after know all requests;
- online: make decisions when requests arrived.

Next:

Offline allocation model: knows everything.

# Problem Formulation of Offline Model

For notational convenience, we use CRT to denote a version of problem, where

- Channel requirement
  - S(single-minded), F(flexible-minded), Y(single channel)
- Region requirement
  - O(overlap), U(unit disks), G(general regions)
- Time requirement
  - I(time interval), D(time duration), M(time interval or duration)

# Example of An Offline Spectrum Allocation Problem

For example, problem SUI represents

- **C**hannel requirement: **S**ingle-minded — all or nothing
- **R**egion requirement: **U**nit Disks
- **T**ime requirement: **I**nterval of a time period.

# Some Well-known Problems

- Problem YOI $\Rightarrow$ maximum weighted independent set problem in interval graphs.
- Problem YOD $\Rightarrow$ knapsack problem.
- Problem YGI with $e_i - s_i \geq T/2$ for each secondary user $i$ $\Rightarrow$ maximum weighted independent set problem of a disk graph
- Problem YUD $\Rightarrow$ multi-knapsack problem is a special case
- problem SOI with $e_i - s_i \geq T/2$ for each secondary user $i$ $\Rightarrow$ set packing problem.

# Some Well-known Problems

- Problem YOI ⇒ maximum weighted independent set problem in interval graphs.
- Problem YOD ⇒ knapsack problem.
- Problem YGI with $e_i - s_i \geq T/2$ for each secondary user $i$ ⇒ maximum weighted independent set problem of a disk graph
- Problem YUD ⇒ multi-knapsack problem is a special case
- problem SOI with $e_i - s_i \geq T/2$ for each secondary user $i$ ⇒ set packing problem.

# Some Well-known Problems

- Problem YOI $\Rightarrow$ maximum weighted independent set problem in interval graphs.
- Problem YOD $\Rightarrow$ knapsack problem.
- Problem YGI with $e_i - s_i \geq T/2$ for each secondary user $i$ $\Rightarrow$ maximum weighted independent set problem of a disk graph
- Problem YUD $\Rightarrow$ multi-knapsack problem is a special case
- problem SOI with $e_i - s_i \geq T/2$ for each secondary user $i$ $\Rightarrow$ set packing problem.

# Some Well-known Problems

- Problem YOI $\Rightarrow$ maximum weighted independent set problem in interval graphs.
- Problem YOD $\Rightarrow$ knapsack problem.
- Problem YGI with $e_i - s_i \geq T/2$ for each secondary user $i$ $\Rightarrow$ maximum weighted independent set problem of a disk graph
- Problem YUD $\Rightarrow$ multi-knapsack problem is a special case
- problem SOI with $e_i - s_i \geq T/2$ for each secondary user $i$ $\Rightarrow$ set packing problem.

# Some Well-known Problems

- Problem YOI $\Rightarrow$ maximum weighted independent set problem in interval graphs.
- Problem YOD $\Rightarrow$ knapsack problem.
- Problem YGI with $e_i - s_i \geq T/2$ for each secondary user $i$ $\Rightarrow$ maximum weighted independent set problem of a disk graph
- Problem YUD $\Rightarrow$ multi-knapsack problem is a special case
- problem SOI with $e_i - s_i \geq T/2$ for each secondary user $i$ $\Rightarrow$ set packing problem.

# Hardness of Offline Allocation

Thus, offline spectrum allocation is NP-hard.

When users required for some subsets of spectrums, no algorithm can achieve ratio $o(\sqrt{m})$ for social efficiency. $m$ is the total number of channels.

# Hardness of Offline Allocation

Thus, offline spectrum allocation is NP-hard.

When users required for some subsets of spectrums, no algorithm can achieve ratio $o(\sqrt{m})$ for social efficiency. $m$ is the total number of channels.

# Our Results: Offline Spectrum Allocation

When we can make decisions offline, we have, for

- problem YOM: 1/2 approximation algorithm
- problem YUI: PTAS
- problem YUD: 1/9 approximation algorithm
- problem YUM: 1/10 approximation algorithm
- problem SUI: $\Theta(\sqrt{m})$ approximation algorithm
- Problem SUD: open
- Problem SUM: solved if SUD solved

# Our Results: Offline Spectrum Auction

What happen if users are selfish?

Based on these methods, we designed truthful mechanisms:

- **Incentive Compatibility**: bidding truthfully is a best choice, regardless of what others do
- **Individual Rationality**: bidding truthfully has non-negative profit

Here we assume that agents only manipulate bid values.

# Our Results: Offline Spectrum Auction

What happen if users are selfish?

Based on these methods, we designed truthful mechanisms:

- **Incentive Compatibility**: bidding truthfully is a best choice, regardless of what others do
- **Individual Rationality**: bidding truthfully has non-negative profit

Here we assume that agents only manipulate bid values.

Offline is almost solved (not completely , ☺);

What will happen if the requests come online and decisions have to be made soon?

⇒ Online Spectrum Allocation and Auction, ☺

Offline is almost solved (not completely , ☺);

What will happen if the requests come online and decisions have to be made soon?

$\Rightarrow$ Online Spectrum Allocation and Auction, ☺

# Online Spectrum Allocation Problem

Challenges:

- what we will get in future?
- should we admit current request(s)?
- what if no spare channels, but we have a bid too good to give up?
- How much should every user be charged?

# Online Spectrum Allocation Problem

Challenges:

- what we will get in future?
- should we admit current request(s)?
- what if no spare channels, but we have a bid too good to give up?
- How much should every user be charged?

# Online Spectrum Allocation Problem

Challenges:

- what we will get in future?
- should we admit current request(s)?
- what if no spare channels, but we have a bid too good to give up?
- How much should every user be charged?

# Online Spectrum Allocation Problem

Challenges:

- what we will get in future?
- should we admit current request(s)?
- what if no spare channels, but we have a bid too good to give up?
- How much should every user be charged?

# Measure Performance of Online Algorithms

## Definition (Competitive Ratio)

$$\varrho(\mathcal{A}) = \min_{\mathcal{I}} \frac{\mathcal{A}(\mathcal{I})}{\text{OPT}(\mathcal{I})}, \text{ where}$$

- $\mathcal{I}$ is *any* possible sequence of requests arrival,
- $\mathcal{A}(\mathcal{I})$ is the profit produced by *online algorithm* $\mathcal{A}$ on $\mathcal{I}$,
- $\text{OPT}(\mathcal{I})$ is the profit produced by optimum *offline algorithm* OPT on $\mathcal{I}$ when $\mathcal{I}$ is known in advance by OPT.

# Nothing is Known About Requests

**Theorem**

*If we know nothing about future requests, we cannot guarantee any competitive ratio of any online spectrum allocation method.*

Adversary model:

$\mathbf{e}_1 = (2, 0, 2)$

$\mathbf{e} = (M, 1, \Delta)$

# Something Must Be Known about Spectrum Requests

In this talk, we assume that one of the following is known

- *time-ratio* $\Delta$: the maximum time requirement by any job is $\Delta$ time slots while the minimum one is 1 time slot, $t_i \in [1, \Delta]$.
- *bid-ratio B*: $B$ is defined as the ratio of maximum bid value to the minimum one, *i.e.*, $D = \max_{i,j} b_i / b_j$.
- *bid density ratio D*: The density $d_i$ of each request $\mathbf{e}_i$ is defined as $\frac{b_i}{t_i}$. $D$ is defined as the ratio of maximum density to minimum one among all jobs, *i.e.*, $D = \max_{i,j} d_i / d_j$.

# Something Must Be Known about Spectrum Requests

In this talk, we assume that one of the following is known

- *time-ratio* $\Delta$: the maximum time requirement by any job is $\Delta$ time slots while the minimum one is 1 time slot, $t_i \in [1, \Delta]$.
- *bid-ratio* $B$: $B$ is defined as the ratio of maximum bid value to the minimum one, *i.e.*, $D = \max_{i,j} b_i/b_j$.
- *bid density ratio* $D$: The density $d_i$ of each request $\mathbf{e}_i$ is defined as $\frac{b_i}{t_i}$. $D$ is defined as the ratio of maximum density to minimum one among all jobs, *i.e.*, $D = \max_{i,j} d_i/d_j$.

# Something Must Be Known about Spectrum Requests

In this talk, we assume that one of the following is known

- *time-ratio* $\Delta$: the maximum time requirement by any job is $\Delta$ time slots while the minimum one is 1 time slot, $t_i \in [1, \Delta]$.
- *bid-ratio* $B$: $B$ is defined as the ratio of maximum bid value to the minimum one, *i.e.*, $D = \max_{i,j} b_i / b_j$.
- *bid density ratio* $D$: The density $d_i$ of each request $\mathbf{e}_i$ is defined as $\frac{b_i}{t_i}$. $D$ is defined as the ratio of maximum density to minimum one among all jobs, *i.e.*, $D = \max_{i,j} d_i / d_j$.

# Preemption or Non-preemption?

If non-preemption, will such known info (time-ratio, bid-ratio, or bid density ratio) be enough for good competitive ratio?

- ✗ known time-ratio $\Delta$,
  $\mathbf{e}_1 = (b_1 = 1, s_1 = 1, t_1 = 2)$, $\mathbf{e}_2 = (b_2 = \infty, s_2 = 2, t_2 = \Delta)$
- bid-ratio $B$
- bid density ratio $D$

# Preemption or Non-preemption?

If non-preemption, will such known info (time-ratio, bid-ratio, or bid density ratio) be enough for good competitive ratio?

- ✗ known time-ratio $\Delta$,
  $\mathbf{e}_1 = (b_1 = 1, s_1 = 1, t_1 = 2)$, $\mathbf{e}_2 = (b_2 = \infty, s_2 = 2, t_2 = \Delta)$
- ✗ bid-ratio $B$
  $\mathbf{e}_1 = (b_1 = 1, s_1 = 1, t_1 = \infty)$,
  $\mathbf{e}_2 = (b_2 = B, s_2 = 2, t_2 = 1), \cdots,$
  $\mathbf{e}_n = (b_n = B, s_n = n, t_n = 1)$
- bid density ratio $D$

# Preemption or Non-preemption?

If non-preemption, will such known info (time-ratio, bid-ratio, or bid density ratio) be enough for good competitive ratio?

- ✗ known time-ratio $\Delta$,

  $\mathbf{e}_1 = (b_1 = 1, s_1 = 1, t_1 = 2)$, $\mathbf{e}_2 = (b_2 = \infty, s_2 = 2, t_2 = \Delta)$

- ✗ bid-ratio $B$

  $\mathbf{e}_1 = (b_1 = 1, s_1 = 1, t_1 = \infty)$,
  $\mathbf{e}_2 = (b_2 = B, s_2 = 2, t_2 = 1), \cdots,$
  $\mathbf{e}_n = (b_n = B, s_n = n, t_n = 1)$

- ✗ bid density ratio $D$

  $\mathbf{e}_1 = (b_1 = 2, s_1 = 1, t_1 = 2)$,
  $\mathbf{e}_2 = (b_2 = D\Delta, s_2 = 2, t_2 = \Delta)$

# Preemption or Non-preemption?

If non-preemption, will such known info (time-ratio, bid-ratio, or bid density ratio) be enough for good competitive ratio?

- ✗ known time-ratio $\Delta$,
  $\mathbf{e}_1 = (b_1 = 1, s_1 = 1, t_1 = 2)$, $\mathbf{e}_2 = (b_2 = \infty, s_2 = 2, t_2 = \Delta)$
- ✗ bid-ratio $B$
  $\mathbf{e}_1 = (b_1 = 1, s_1 = 1, t_1 = \infty)$,
  $\mathbf{e}_2 = (b_2 = B, s_2 = 2, t_2 = 1), \cdots,$
  $\mathbf{e}_n = (b_n = B, s_n = n, t_n = 1)$
- ✗ bid density ratio $D$
  $\mathbf{e}_1 = (b_1 = 2, s_1 = 1, t_1 = 2)$,
  $\mathbf{e}_2 = (b_2 = D\Delta, s_2 = 2, t_2 = \Delta)$

Thus, without preemption, the competitive ratio of any online method could be arbitrarily bad (even know some other info).

# Need good online method?

To get bounded worst case competitive ratio, we need

1. Preemption, and
2. some other additional info about time-ratio, bid-ratio, or bid-density ratio

Is Preemption Free?

# Need good online method?

To get bounded worst case competitive ratio, we need

1. Preemption, and
2. some other additional info about time-ratio, bid-ratio, or bid-density ratio

Is Preemption Free?

# Need good online method?

To get bounded worst case competitive ratio, we need

1. Preemption, and
2. some other additional info about time-ratio, bid-ratio, or bid-density ratio

Is Preemption Free?

- ✓Free: admit the best current choice (simple, ☺)
- Not-Free: need pay penalty for preempting a spectrum usage.

# Need good online method?

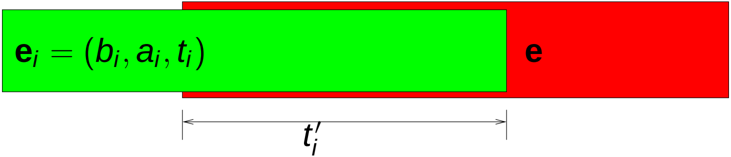
To get bounded worst case competitive ratio, we need

1. Preemption, and
2. some other additional info about time-ratio, bid-ratio, or bid-density ratio

Is Preemption Free?

- ✓Free: admit the best current choice (simple, ☺)
- Not-Free: need pay penalty for preempting a spectrum usage.

# How much should we compensate?



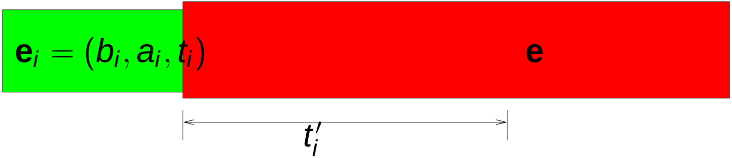

First assume that the preempted spectrum usage is compensated with

$$\gamma(b_i, t_i, t_i') = \beta \frac{t_i'}{t_i} b_i$$

for a constant $\beta > 0$. Here

- $t_i'$ is the unserved timeslots of user $i$
- $t_i$ is the requested timeslots of user $i$
- $b_i$ is the bid by $i$

# How much should we compensate?




First assume that the preempted spectrum usage is compensated with

$$\gamma(b_i, t_i, t_i') = \beta \frac{t_i'}{t_i} b_i$$

for a constant $\beta > 0$. Here

- $t_i'$ is the unserved timeslots of user $i$
- $t_i$ is the requested timeslots of user $i$
- $b_i$ is the bid by $i$

# Efficient Methods and Upper bounds

Assume that we know the time-ratio $\Delta$ only

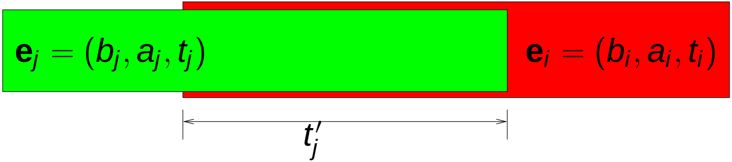
- $0 \leq \beta < 1$
- $\beta > 1$
- $\beta = 1$

# Efficient Methods and Upper bounds

Assume that we know the time-ratio $\Delta$ only

- $0 \leq \beta < 1$
- $\beta > 1$
- $\beta = 1$

# $0 \leq \beta < 1$: Simple Greedy

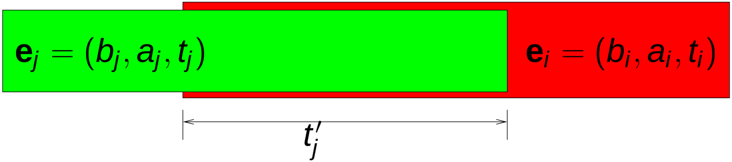



**Greedy Algorithm $\mathcal{G}_{<1}$**

1: Select the request $\mathbf{e}_i$ which has the largest bid among all coming requests that arrive at time $t$.

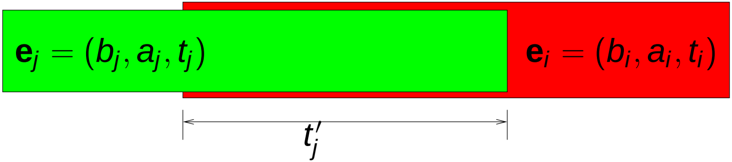
2: If channel is empty, satisfy $\mathbf{e}_i$; Otherwise, satisfy $\mathbf{e}_i$ by preempting current request.

# It is Competitive



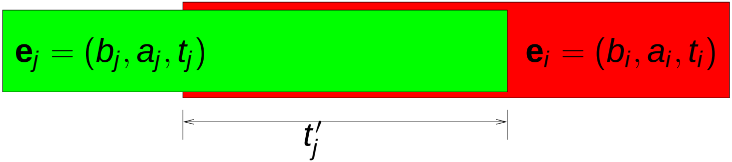

- At each time $a_j$, algorithm $\mathcal{G}_{<1}$ makes at least $(1-\beta)b_j$ profit.
- On the other hand, the optimal algorithm makes at most $b_j$ at each time $a_j$ since $b_j$ is the largest bid at that time.
- Therefore, $\mathcal{G}_{<1}$ is $(1-\beta)$-competitive.

# It is Competitive




- At each time $a_j$, algorithm $\mathcal{G}_{<1}$ makes at least $(1-\beta)b_j$ profit.
- On the other hand, the optimal algorithm makes at most $b_j$ at each time $a_j$ since $b_j$ is the largest bid at that time.
- Therefore, $\mathcal{G}_{<1}$ is $(1-\beta)$-competitive.

# It is Competitive



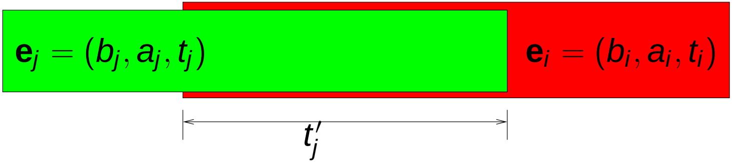

- At each time $a_j$, algorithm $\mathcal{G}_{<1}$ makes at least $(1-\beta)b_j$ profit.
- On the other hand, the optimal algorithm makes at most $b_j$ at each time $a_j$ since $b_j$ is the largest bid at that time.
- Therefore, $\mathcal{G}_{<1}$ is $(1-\beta)$-competitive.

# Efficient Methods and Upper bounds

Assume that we know the time-ratio $\Delta$ only

- $0 \leq \beta < 1$
- $\beta > 1$
- $\beta = 1$

# Efficient Online Spectrum Allocation with $\beta > 1$




**Greedy Algorithm $\mathcal{G}_{>1}$**

1: If the channel is empty, $\mathbf{e}_i$ will be satisfied anyway.

2: If channel is being used by a request $\mathbf{e}_j$, $\mathbf{e}_i$ preempts $\mathbf{e}_j$ only if

$$b_i \geq 2\gamma(b_j, t_j, t'_j)$$

# Competitive Ratio of $\mathcal{G}_{>1}$ when $\beta > 1$

**Theorem**

*Algorithm* $\mathcal{G}_{>1}$ *is* $\frac{1}{4\beta}\Delta^{-1}$*-competitive.*

When $\beta > 1$, what is the upper bound on competitive ratio of any deterministic method?

# Performance Upper Bound

## Theorem

*There is no online algorithm with competitive ratio more than* $\frac{2\beta}{(\beta-1)^2}\Delta^{-1}$ *when* $\beta > 1$*.*

Recall that our algorithm $\mathcal{G}_{>1}$ has competitive ratio

$$\frac{1}{4\beta}\Delta^{-1}.$$

# Performance Upper Bound ($\beta > 1$)

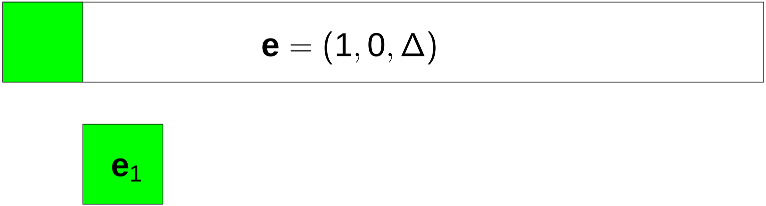



Figure: Requests arrive at/before time 1

$\mathbf{e}_1 = (\beta^{\frac{\Delta-1}{\Delta}} - 1, 1, 1)$.
Central Authority has to accept **e** and reject $\mathbf{e}_1$.

# Performance Upper Bound ($\beta > 1$)

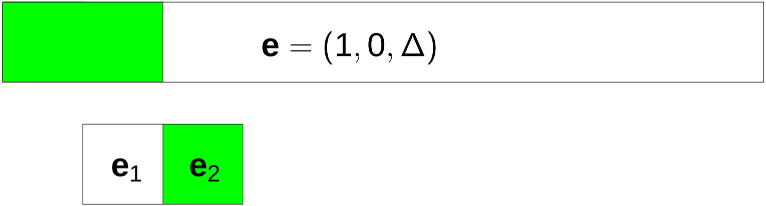



Figure: Requests arrive at/before time 2

$\mathbf{e}_2 = (\beta^{\frac{\Delta - 2}{\Delta}} - 1, 2, 1)$.
Central Authority has to reject $\mathbf{e}_2$.

# Performance Upper Bound ($\beta > 1$)

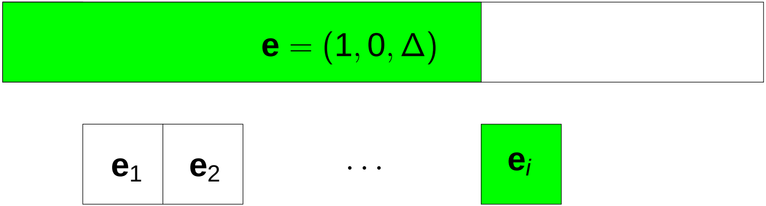



Figure: Requests arrive at/before time $i$

$\mathbf{e}_i = (\beta^{\frac{\Delta - i}{\Delta}} - 1, i, 1)$.
Central Authority has to reject $\mathbf{e}_i$.

# Performance Upper Bound ($\beta > 1$)

$$\mathbf{e} = (1, 0, \Delta)$$

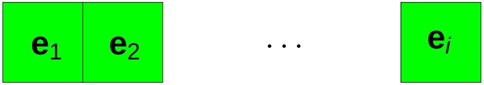



Figure: Requests arrive at/before time $i$

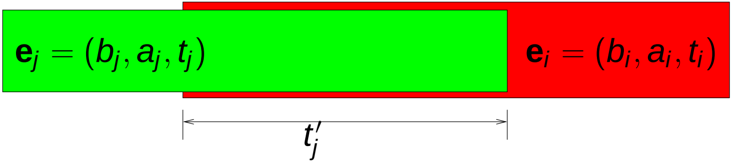
Any online algorithm makes at most 1 profit, while the optimal *offline* algorithm makes $\sum_{i=0}^{\frac{\beta-1}{\beta}\Delta} \beta^{\frac{\Delta - i}{\Delta}} - 1 = \frac{(\beta-1)^2}{2\beta}\Delta - \frac{\beta-1}{2}$.
The competitive ratio is no more than $\frac{2\beta}{(\beta-1)^2}\Delta^{-1}$.

# Efficient Methods and Upper bounds

Assume that we know the time-ratio $\Delta$ only

- $0 \leq \beta < 1$
- $\beta > 1$
- $\beta = 1$

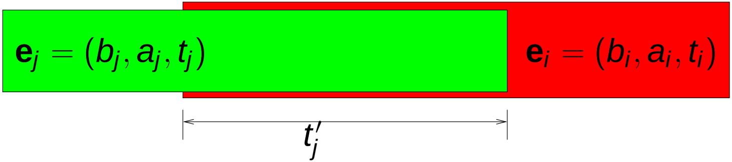
# What to preempt?




Two possible preemption scenarios:

- the bid of new request is much larger;
- even the new request is later preempted after only one time slot, the profit made is not small, compared with its bid and bids of preempted requests.

# What to preempt?




Two possible preemption scenarios:

- the bid of new request is much larger;
- even the new request is later preempted after only one time slot, the profit made is not small, compared with its bid and bids of preempted requests.

# Efficient Online Spectrum Allocation with $\beta = 1$

**Greedy Algorithm $\mathcal{G}_{\mathcal{T}}$ With Constant $c > 1$**

1: If the channel is empty, $\mathbf{e}_i$ will be satisfied anyway.
2: If the channel is being used by other request $\mathbf{e}_j$, $\mathbf{e}_i$ preempts $\mathbf{e}_j$ if and only if

$$\begin{cases} \textbf{Strong Preemption:} & b_i \geq c \cdot b_j \text{ or} \\ \textbf{Weak Preemption:} & \frac{b_i}{t_i} + \left(b_j - \gamma(b_j, t_j, t_j')\right) > \Delta^{-\frac{1}{2}} b_j \end{cases}$$

# Performance Analysis

**Theorem**

*Algorithm $\mathcal{G}_{\mathcal{T}}$ is $\frac{c-1}{2c(c+2)}\Delta^{-\frac{1}{2}}$-competitive.*

When $c = 1 + \sqrt{3}$, competitive ratio is maximized at

$$\frac{\sqrt{3}}{12 + 8\sqrt{3}}\Delta^{-\frac{1}{2}}.$$

When $\beta = 1$, what is the upper bound on competitive ratio of any deterministic method?

# Performance Upper Bound: First Try

## Theorem

*There is no online algorithm with competitive ratio more than* $2\Delta^{-\frac{1}{3}}$.

We will prove by adversary model.

# Performance Upper Bound ($\beta = 1$): First Try

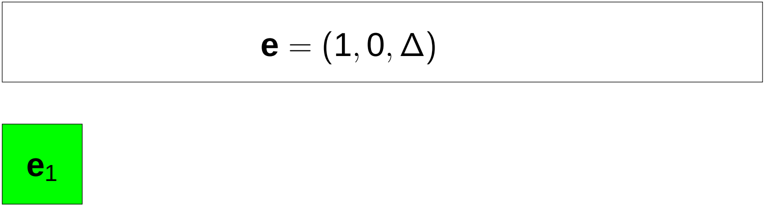



Figure: Requests arrive at/before time 0

$\mathbf{e}_1 = (2\Delta^{-\frac{1}{3}}, 0, 1)$.
Central Authority has to accept $\mathbf{e}$ and reject $\mathbf{e}_1$.

# Performance Upper Bound ($\beta = 1$): First Try

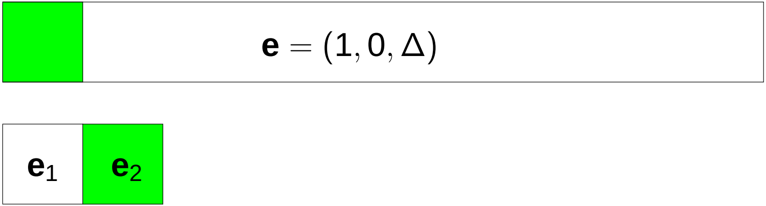



Figure: Requests arrive at/before time 1

$\mathbf{e}_2 = (2\Delta^{-\frac{1}{3}} - \frac{1}{\Delta}, 1, 1)$.
Central Authority has to reject $\mathbf{e}_2$.

# Performance Upper Bound ($\beta = 1$): First Try

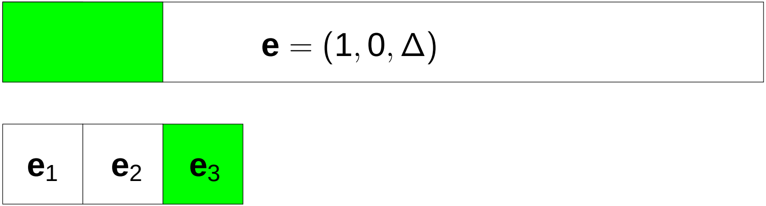



Figure: Requests arrive at/before time 2

$\mathbf{e}_3 = (2\Delta^{-\frac{1}{3}} - \frac{2}{\Delta}, 2, 1)$.

Central Authority has to reject $\mathbf{e}_3$.

# Performance Upper Bound ($\beta = 1$): First Try

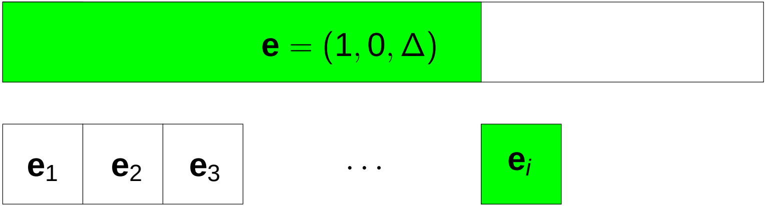



Figure: Requests arrive at/before time $i - 1$

$\mathbf{e}_i = (2\Delta^{-\frac{1}{3}} - \frac{i-1}{\Delta}, i - 1, 1)$.
Central Authority has to reject $\mathbf{e}_i$.

# Performance Upper Bound ($\beta = 1$): First Try

$$\mathbf{e} = (1, 0, \Delta)$$

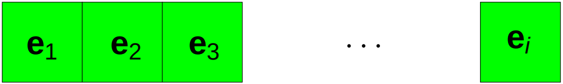



Figure: Requests arrive at/before time $i - 1$

Any online algorithm makes at most 1 profit, while the optimal *offline* algorithm makes $\sum_{i=0}^{2\Delta^{\frac{2}{3}}} (2\Delta^{-\frac{1}{3}} - \frac{i}{\Delta}) = 2\Delta^{\frac{1}{3}} + \Delta^{-\frac{1}{3}}$. The competitive ratio is no more than $2\Delta^{-\frac{1}{3}}$.

# Performance Upper Bound ($\beta = 1$): Second Try

**Theorem**

*There is no online algorithm with competitive ratio more than* $\frac{1}{\rho}$ *for constant* $\rho$ *with* $\frac{1}{2}\Delta^{\frac{1}{3}} \leq \rho < \frac{\sqrt{2}}{2}\Delta^{\frac{1}{2}}$.

Thus, the best competitive ratio, for $\beta = 1$, is

$$\sqrt{2}\Delta^{-\frac{1}{2}}$$

Recall that, our algorithm achieved

$$\frac{\sqrt{3}}{12 + 8\sqrt{3}}\Delta^{-\frac{1}{2}}.$$

We will prove by adversary model.

# Performance Upper Bound ($\beta = 1$): Second Try

**Theorem**

*There is no online algorithm with competitive ratio more than* $\frac{1}{\rho}$ *for constant* $\rho$ *with* $\frac{1}{2}\Delta^{\frac{1}{3}} \leq \rho < \frac{\sqrt{2}}{2}\Delta^{\frac{1}{2}}$.

Thus, the best competitive ratio, for $\beta = 1$, is

$$\sqrt{2}\Delta^{-\frac{1}{2}}$$

Recall that, our algorithm achieved

$$\frac{\sqrt{3}}{12 + 8\sqrt{3}}\Delta^{-\frac{1}{2}}.$$

We will prove by adversary model.

# Performance Upper Bound ($\beta = 1$): Second Try

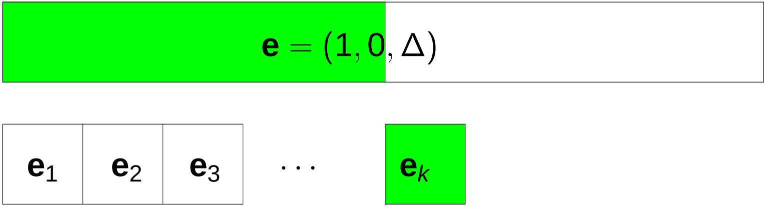



Figure: Requests arrive at/before time $k - 1$

Similarly, all online algorithm will reject $\mathbf{e}_i = (2\Delta^{-\frac{1}{3}} - \frac{i-1}{\Delta}, i - 1, 1)$ for $i = 1, \cdots, k$.
Here $k$ is the smallest integer such that $\sum_{i=1}^{k} b_i = S_k > 1$.

# Performance Upper Bound ($\beta = 1$): Second Try

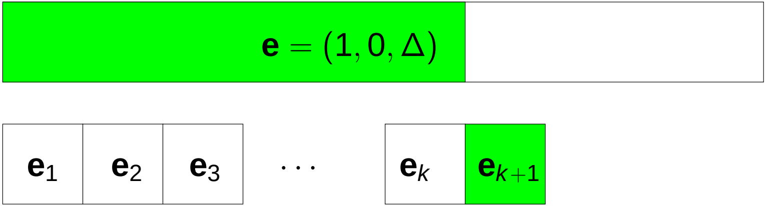



Figure: Requests arrive at/before time $k$

Central authority has to reject $\mathbf{e}_{k+1}$ while $b_{k+1} + \frac{k}{\Delta} < \frac{1}{\rho}(S_k + b_{k+1})$.

$b_{k+1} = \frac{1}{\Delta} + \frac{\sqrt{\Delta^2 - 2\rho^2\Delta}}{\rho\Delta}$ is a feasible value.

# Performance Upper Bound ($\beta = 1$): Second Try

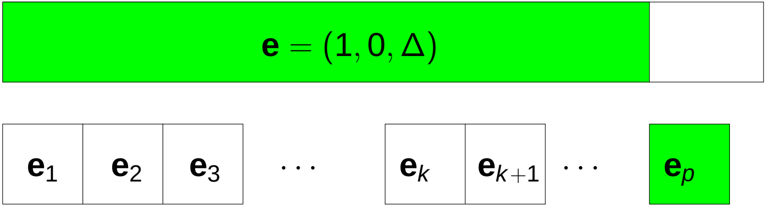



Figure: Requests arrive at/before time $p - 1$

Central authority has to reject $\mathbf{e}_p$ while $b_p + \frac{p-1}{\Delta} < \frac{1}{\rho}(S_k + \sum_{i=k+1}^{p-1} b_i)$.

It can be proved by induction that $b_i \geq b_{k+1} = \frac{1}{\Delta} + \frac{\sqrt{\Delta^2 - 2\rho^2\Delta}}{\rho\Delta}$ for all $i \geq k+1$.

# Performance Upper Bound: Second Try

$$\mathbf{e} = (1, 0, \Delta)$$

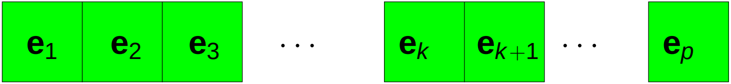



Figure: Requests arrive at/before time $p - 1$

Any online algorithm makes at most 1 profit, while the optimal *offline* algorithm makes at least $(\Delta - k)b_{k+1} \geq (\Delta - k)(\frac{1}{\Delta} + \frac{\sqrt{\Delta^2 - 2\rho^2\Delta}}{\rho\Delta})$.

This profit is always larger than $\rho$ when $\frac{1}{2}\Delta^{\frac{1}{3}} \leq \rho < \frac{\sqrt{2}}{2}\Delta^{\frac{1}{2}}$.

If we know some other information instead?

- Know the bid ratio $B$ only: design asymptotically optimum method;
- Know the bid density ratio $D$ only: design asymptotically optimum method;

If we know some other information instead?

- Know the bid ratio $B$ only: design asymptotically optimum method;
- Know the bid density ratio $D$ only: design asymptotically optimum method;

# With other known info: Bid Ratio $B$

When $\beta = 1$,

1. Designed a method with competitive ratio $\frac{1}{1+B}$.
2. Showed that no method guarantees competitive ratio $> \frac{1}{B}$.

When $\beta > 1$, no method can guarantee any ratio.

# With other known info: Bid Ratio $B$

When $\beta = 1$,

1. Designed a method with competitive ratio $\frac{1}{1+B}$.
2. Showed that no method guarantees competitive ratio $> \frac{1}{B}$.

When $\beta > 1$, no method can guarantee any ratio.

# With other known info: Bid Density Ratio $D$

When $\beta = 1$,

1. Designed a method with competitive ratio $\frac{1}{1+D}$.
2. Showed that no method guarantees competitive ratio $> \frac{2}{D}$.

# With other known info: Bid Density Ratio $D$

When $\beta > 1$,

1. Designed a method with competitive ratio $\frac{1}{2(\beta+D)}$.
2. Showed that no method guarantees competitive ratio $> \frac{2\beta}{(\beta-1)D}$.

# More General Penalty Function

We also study the case when the penalty function

$$\gamma(b_i, t_i, t_i') = (\alpha + \beta \frac{t_i'}{t_i}) b_i$$

We designed asymptotically optimum methods for

1. $\alpha + \beta < 1$
2. $\alpha + \beta = 1$
3. $\alpha + \beta > 1$

# Experiments Setup

In our experiment, we generate random requests with random bid, time or density requirements.

1. uniformly pick $t_i \in [1, \Delta]$
2. bid $b_i \in [1, 10000]$.

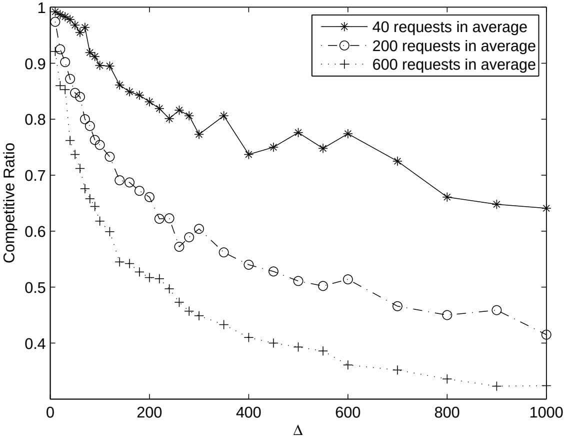
For algorithm $\mathcal{G}_{\mathcal{B}}$,

1. bid randomly in $[1, B]$, and time $t_i \in [1, 1000]$.

For algorithm $\mathcal{G}_{\mathcal{D}}$,

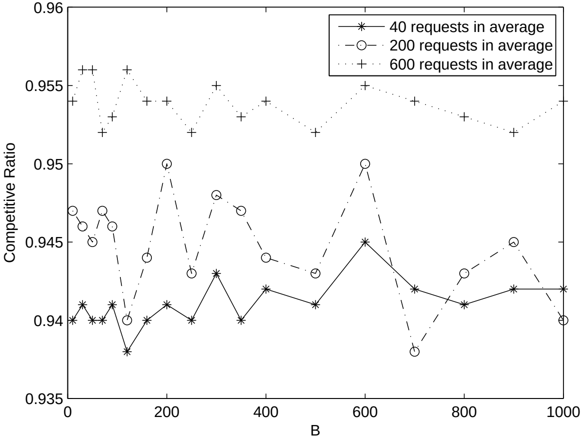
1. bid randomly in $[1, \lfloor\sqrt{D}\rfloor]$ and time $t_i \in [1, \lfloor\sqrt{D}\rfloor]$.

# The Achieved Competitive Ratio $\mathcal{G}_{\mathcal{T}}$



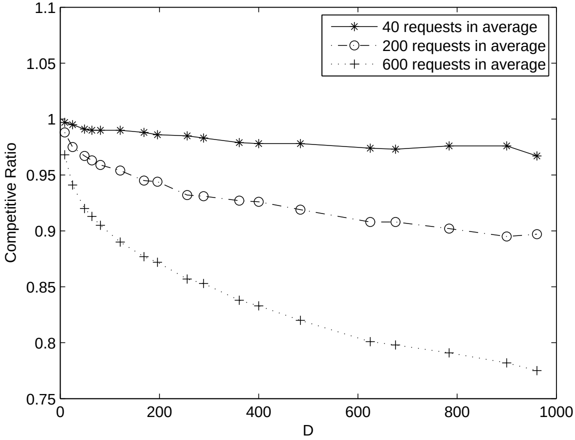

(a) Algorithm $\mathcal{G}_{\mathcal{T}}$,

# The Achieved Competitive Ratio $\mathcal{G}_{\mathcal{B}}$




(b) Algorithm $\mathcal{G}_{\mathcal{B}}$

# The Achieved Competitive Ratio $\mathcal{G}_{\mathcal{D}}$




(c) Algorithm $\mathcal{G}_{\mathcal{D}}$

# Non-instantaneous Requests

So far, we assumed that

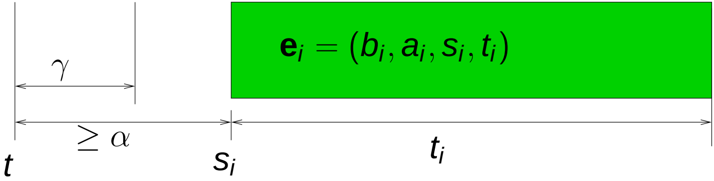
- Each request arrived at time $a_i$, asked for $t_i$ timeslots starting from $a_i$.
- The request thus must be immediately processed.

# Non-instantaneous Requests

In rest of talk, we assume that

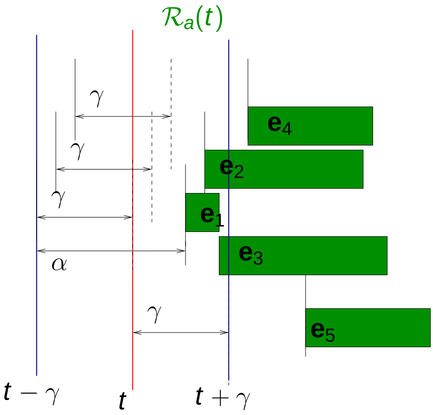
- Each request arrived at time $a_i$, asked for $t_i$ timeslots starting from $s_i$ with $s_i \geq a_i + \alpha$.
- The request should be processed within $\gamma$.

We call it $(\beta, \alpha, \gamma)$ problem.



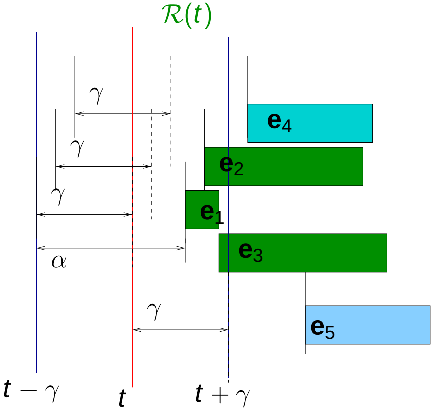
# What do we know at time $t$?

All requests $\mathcal{R}_a(t)$



# What do we know at time $t$?

All requests $\mathcal{R}(t) \subseteq \mathcal{R}_a(t)$



# Candidate Requests Set

## Definition (Strong Candidate Requests Set)

A *strong candidate requests set* at time $t$, denoted as $\mathcal{C}_1(t)$, is a subset of requests from $\mathcal{R}(t)$ that has the largest total bids if $\mathcal{C}_1(t)$ is allowed to run without preemption, from time $t-\gamma+\alpha$ to timeslots at most $t+\alpha+\Delta$.

For set $\mathcal{C}_1(t)$, let $\mathcal{P}(\mathcal{C}_1(t), t')$ denote the profit made from $\mathcal{C}_1(t)$ if these requests are admitted and then possibly being preempted at a time-slot $t'$.

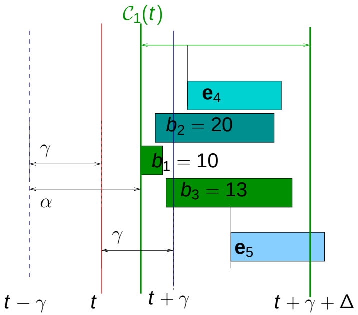
# Candidate Requests Set

**Definition (Weak Candidate Requests Set)**

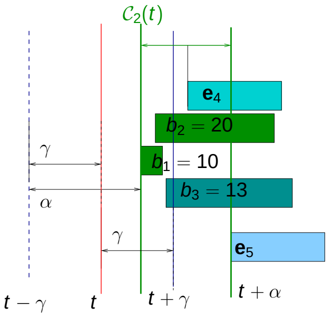
A *weak candidate requests set* at time $t$, denoted as $\mathcal{C}_2(t)$, is a subset of requests from $\mathcal{R}(t)$ that has the largest total bids if $\mathcal{C}_2(t)$ is allowed to run during time interval $[t-\gamma+\alpha, t+\alpha]$ (thus, these requests may be preempted by some requests started on time-slot $t+\alpha+1$).

# Candidate Requests Sets

candidate requests sets $\mathcal{C}_1(t), \mathcal{C}_2(t) \subseteq \mathcal{R}(t)$




Strong $\mathcal{C}_1(t) \subseteq \mathcal{R}(t)$




Weak $\mathcal{C}_2(t) \subseteq \mathcal{R}(t)$

# Efficient Methods

1. for $\beta = 1$
2. for $\beta > 1$

# Efficient Methods

1. for $\beta = 1$
2. for $\beta > 1$

# Efficient Methods for $\beta = 1$

**Input**: A constant parameter $c_1 > 1$, an adjustable control parameter $c_2 > 0$, $\mathcal{C}_1(t)$, and $\mathcal{C}_2(t)$.

*Current candidate requests set* $\mathcal{C}$ from time $t' < t$. Here $\mathcal{C} = \mathcal{C}_1(t')$ if $\mathcal{C}_1(t')$ strongly preempted others, or $\mathcal{C} = \mathcal{C}_2(t')$ if $\mathcal{C}_2(t')$ strongly preempted others.

**Output**: new *current candidate requests set* $\mathcal{C}$.

# Efficient Methods for $\beta = 1$

**if** $\mathcal{C} = \mathcal{C}_2(t')$ **then**
    **if** $t - t' \geq \gamma$ **then**
        $\mathcal{C} = \emptyset$;
    **else**
        Accept earliest request $\mathbf{e}_i \in \mathcal{C}_2(t)$
**if** $\mathcal{C} = \mathcal{C}_1(t')$ or $\emptyset$ **then**
    **if** $\mathcal{C}_1(t) \geq c_1 \cdot \mathcal{C}_1(t')$ **then**
        $\mathcal{C} = \mathcal{C}_1(t)$; Accept earliest request $\mathbf{e}_i \in \mathcal{C}_1(t)$
    **else if** $\mathcal{C}_2(t) + \mathcal{P}(\mathcal{C}_1(t'), t) \geq c_2 \cdot \mathcal{C}_1(t)$ **then**
        $\mathcal{C} = \mathcal{C}_2(t)$; Accept earliest request $\mathbf{e}_i \in \mathcal{C}_2(t)$
    **else**
        Accept request $\mathbf{e}_i \in \mathcal{C}_1(t')$ such that $s_i = t - \gamma + \alpha$.

# Competitive Ratio when $\beta = 1$

**Theorem**

*Algorithm $\mathcal{G}$ is $\Theta(\sqrt{\gamma + 1}\Delta^{-\frac{1}{2}})$-competitive when $\gamma = O(\Delta)$.*

# Efficient Methods

1. for $\beta = 1$
2. for $\beta > 1$

# Efficient Online Method $\mathcal{H}$ when $\beta > 1$

**Input**: A constant parameter $c > 1+\beta$, $\gamma$, $\alpha$, $\Delta$, $\mathcal{R}_a(t)$, $\mathcal{R}(t)$, $\mathcal{C}_1(t)$. Previous *current candidate requests set* $\mathcal{C} = \mathcal{C}_1(t')$ where $t' < t$. Here $\mathcal{C}_1(t')$ may be empty.
**Output**: whether requests submitted at time $t-\gamma$ will be admitted and new *current candidate requests set* $\mathcal{C}$.

1: **if** $\mathcal{C}_1(t) \geq c \cdot \mathcal{C}_1(t')$ **then**
2: $\quad \mathcal{C} = \mathcal{C}_1(t)$;
3: $\quad$ Accept request $\mathbf{e}_i \in \mathcal{C}_1(t)$ such that $a_i = t-\gamma$.
4: **else**
5: $\quad$ Accept request $\mathbf{e}_i \in \mathcal{C}_1(t')$ such that $a_i = t-\gamma$.

# Competitive Ratio

**Theorem**

*Algorithm $\mathcal{H}$ is $\frac{(c-\beta-1)}{c^2}\frac{\gamma+1}{\Delta+\gamma+1}$ competitive.*

When $\gamma = a\Delta - 1$, it is easy to show that

**Theorem**

*Method $\mathcal{H}$ has a competitive ratio at least, by choosing $c = 2(1+\beta)$,*

$$\frac{a}{4(1+a)(1+\beta)}.$$

# Upper bounds

What is the best we can do?

1. for $\beta = 1$
2. for $\beta > 1$

# Performance Upper Bound ($\beta = 1$): First Try

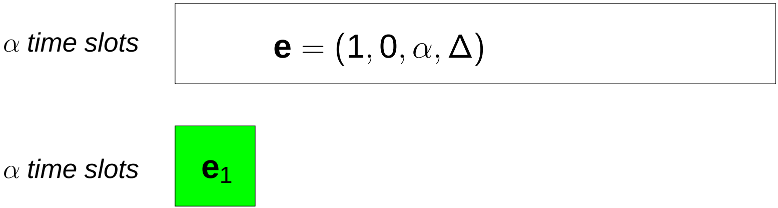



Figure: Requests arrive at/before time 0

$\mathbf{e}_1 = (\sqrt[3]{2(\gamma + 1)}\Delta^{-\frac{1}{3}}, 0, \alpha, 1)$.
Central Authority has to accept **e** and reject $\mathbf{e}_1$.

# Performance Upper Bound ($\beta = 1$): First Try

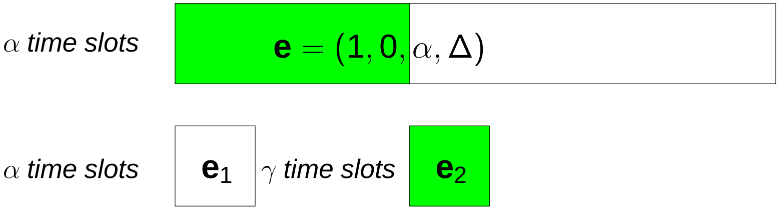



Figure: Requests arrive at/before time $\gamma + 1$

$\mathbf{e}_2 = (\sqrt[3]{2(\gamma+1)}\Delta^{-\frac{1}{3}} - \frac{\gamma+1}{\Delta}, \gamma+1, \gamma+1+\alpha, 1)$.
Central Authority has to reject $\mathbf{e}_2$.

# Performance Upper Bound ($\beta = 1$): First Try

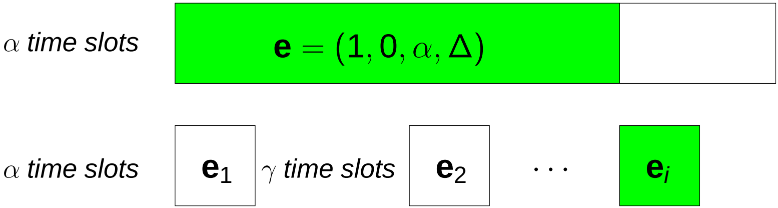



Figure: Requests arrive at/before time $(i-1)(\gamma+1)$

$\mathbf{e}_i = (\sqrt[3]{2(\gamma+1)}\Delta^{-\frac{1}{3}} - \frac{(i-1)(\gamma+1)}{\Delta}, (i-1)\gamma+1, (i-1)(\gamma+1)+\alpha, 1)$.
Central Authority has to reject $\mathbf{e}_i$.

# Performance Upper Bound ($\beta = 1$): First Try

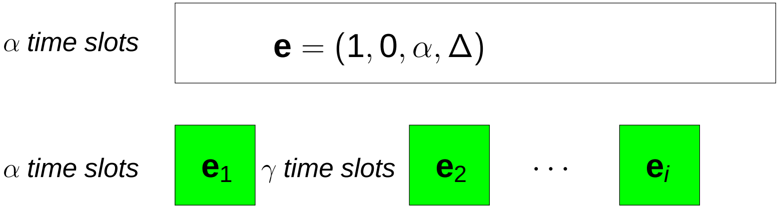



Figure: Requests arrive at/before time $(i-1)(\gamma+1)$

Any online algorithm makes at most 1 profit, while the optimal *offline* algorithm makes $\sum_{i=1}^{n} b_i \geq \frac{\Delta^{\frac{1}{3}}}{\sqrt[3]{2(\gamma+1)}}$ profit ($b_n > 0$).

The competitive ratio is no more than $\sqrt[3]{2(\gamma+1)}\Delta^{-\frac{1}{3}}$.

# Performance Upper Bound ($\beta = 1$): Second Try

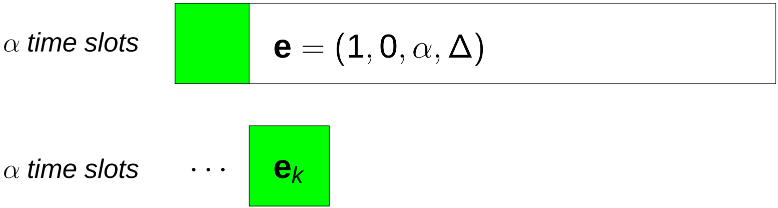



Figure: Requests arrive at/before time $(k-1)(\gamma+1)$

Similarly, all online algorithm will reject
$\mathbf{e}_i = (\sqrt[3]{2(\gamma+1)}\Delta^{-\frac{1}{3}} - \frac{(i-1)(\gamma+1)}{\Delta}, (i-1)\gamma+1, (i-1)(\gamma+1)+\alpha, 1)$
for $i = 1, \cdots, k$.
Here $k$ is the smallest integer such that $\sum_{i=1}^{k} b_i = S_k > 1$.

# Performance Upper Bound ($\beta = 1$): Second Try

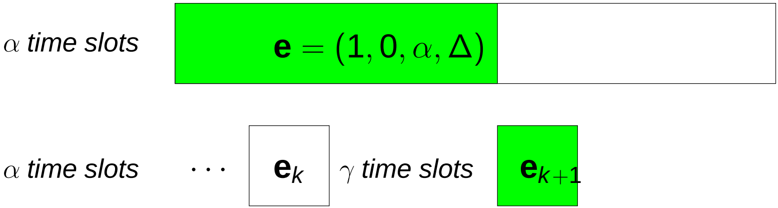



Figure: Requests arrive at/before time $k(\gamma + 1)$

Central authority has to reject $\mathbf{e}_{k+1}$ while

$$b_{k+1} + \frac{k(\gamma+1)}{\Delta} < \frac{1}{c}(S_k + b_{k+1}).$$

$b_{k+1} = \frac{1}{\Delta} + \frac{\sqrt{\Delta^2 - 2(\gamma+1)c^2\Delta}}{c\Delta}$ is a feasible value.

# Performance Upper Bound ($\beta = 1$): Second Try

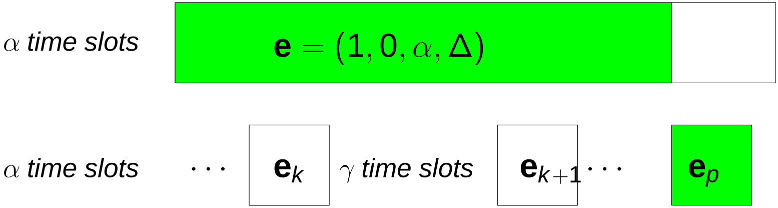



Figure: Requests arrive at/before time $(p-1)(\gamma+1)$

Central authority has to reject $\mathbf{e}_p$ while
$b_p + \frac{(p-1)(\gamma+1)}{\Delta} < \frac{1}{c}(S_k + \sum_{i=k+1}^{p-1} b_i)$.
It can be proved by induction that
$b_i \geq b_{k+1} = \frac{1}{\Delta} + \frac{\sqrt{\Delta^2 - 2(\gamma+1)c^2\Delta}}{c\Delta}$ for all $i \geq k+1$.

# Performance Upper Bound ($\beta = 1$): Second Try

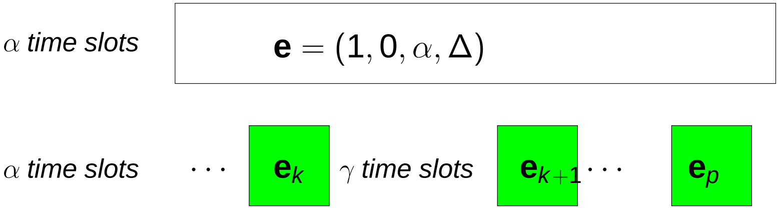



Figure: Requests arrive at/before time $(p-1)(\gamma+1)$

Any online algorithm makes at most 1 profit, while the optimal *offline* algorithm makes at least
$(\Delta - k)b_{k+1} \geq (\Delta - k)(\frac{1}{\Delta} + \frac{\sqrt{\Delta^2 - 2(\gamma+1)c^2\Delta}}{c\Delta})$.
This profit is always larger than $c$ when
$\sqrt{2(\gamma+1)}\Delta^{-\frac{1}{2}} < \frac{1}{c} \leq \sqrt[3]{2(\gamma+1)}\Delta^{-\frac{1}{3}}$.

# Upper bounds

What is the best we can do?

1. for $\beta = 1$
2. for $\beta > 1$

# Performance Upper Bound ($\beta > 1$)

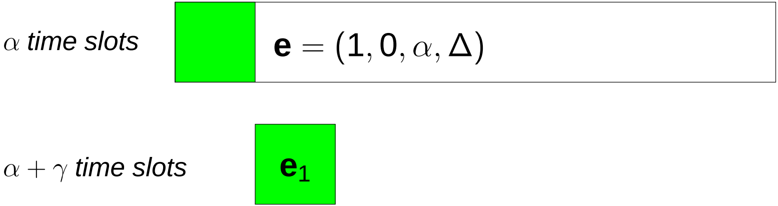



Figure: Requests arrive at/before time $\gamma + 1$

$\mathbf{e}_1 = (\beta - 1 - \frac{\beta(\gamma+1)}{\Delta}, \gamma + 1, \gamma + 1 + \alpha, 1)$.
Central Authority has to accept $\mathbf{e}$ and reject $\mathbf{e}_1$.

# Performance Upper Bound ($\beta > 1$)

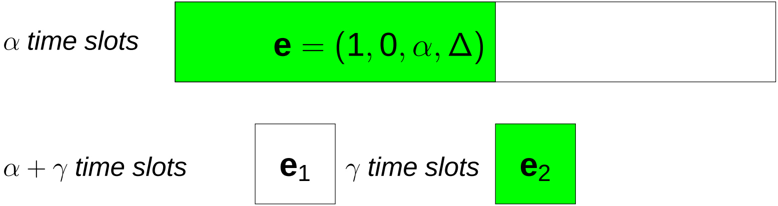



Figure: Requests arrive at/before time $2(\gamma + 1)$

$\mathbf{e}_2 = (\beta - 1 - \frac{2\beta(\gamma+1)}{\Delta}, 2(\gamma + 1), 2(\gamma + 1) + \alpha, 1)$.
Central Authority has to reject $\mathbf{e}_2$.

# Performance Upper Bound ($\beta > 1$)

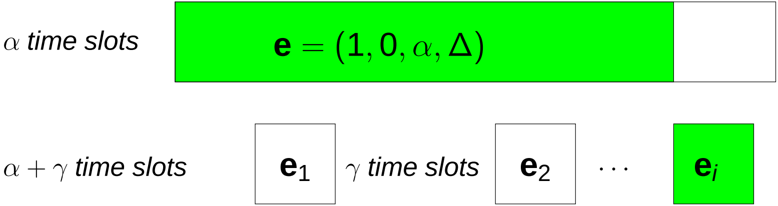



Figure: Requests arrive at/before time $i(\gamma + 1)$

$\mathbf{e}_i = (\beta - 1 - \frac{i\beta(\gamma+1)}{\Delta}, i(\gamma + 1), i(\gamma + 1) + \alpha, 1)$.
Central Authority has to reject $\mathbf{e}_i$.

# Performance Upper Bound ($\beta > 1$)

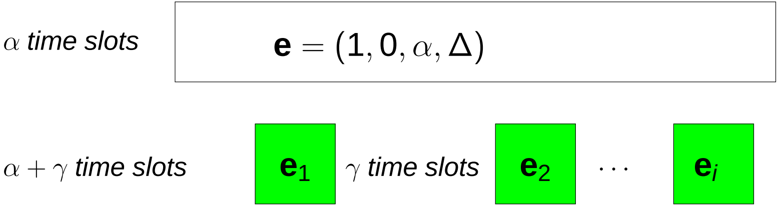



Figure: Requests arrive at/before time $i(\gamma + 1)$

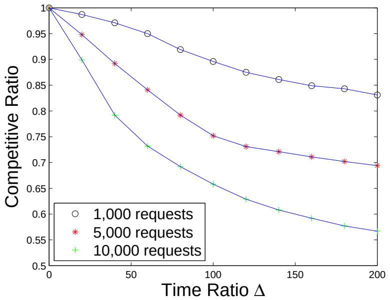
Any online algorithm makes at most 1 profit, while the optimal *offline* algorithm makes $\sum_{i=1}^{n} b_i \geq \frac{(\beta-1)^2}{2\beta(\gamma+1)}\Delta$ ($b_n > 0$).

The competitive ratio is no more than $\frac{2\beta(\gamma+1)}{(\beta-1)^2}\Delta^{-1}$.

# Simulation Studies

1. Competitive Ratios
2. Efficiency ratio
3. Spectrum utilization
4. Compared with Simple Greedy Methods

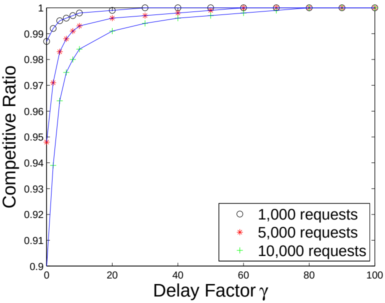
# Competitive ratios



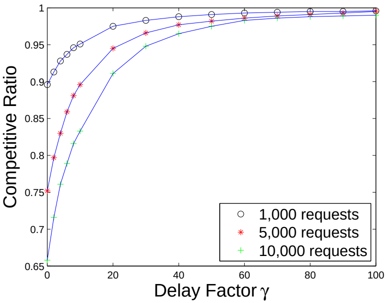

(a) *Delay Factor* $\gamma = 0$, (b) *Delay Factor* $\gamma = 20$,

Figure: The competitive ratios of method $\mathcal{G}$ in various cases.

# Competitive ratios



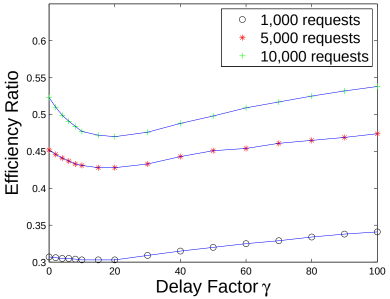

(c) *Time Ratio* $\Delta = 20$,



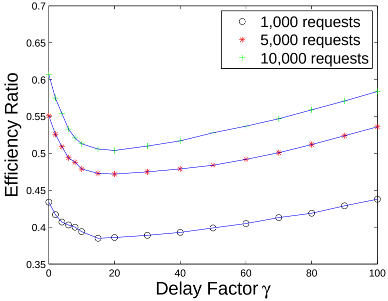

(d) *Time Ratio* $\Delta = 100$.

Figure: The competitive ratios of method $\mathcal{G}$ in various cases.

# Efficiency Ratios



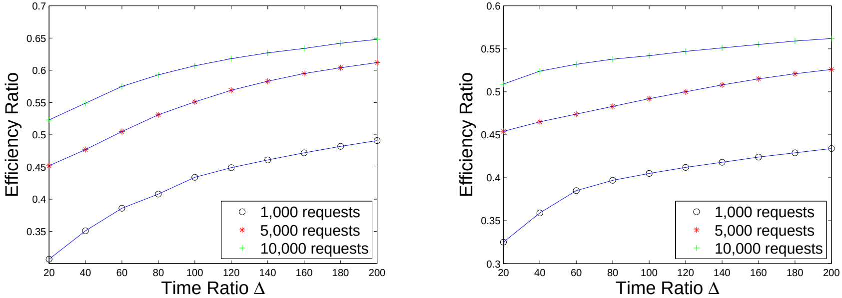

(a) *Time Ratio* $\Delta = 20$,




(b) *Time Ratio* $\Delta = 100$,

Figure: The efficiency ratios of our mechanism in various cases.

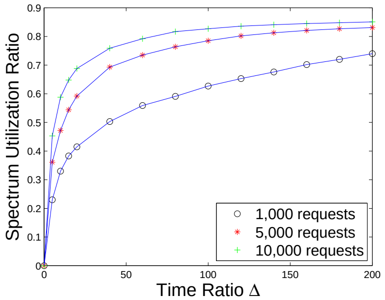
# Efficiency Ratios



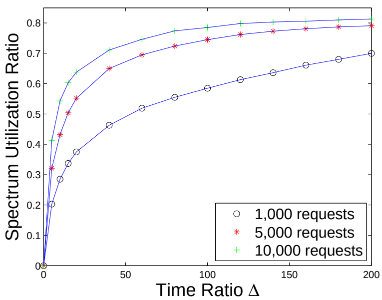

(c) *Delay Factor* $\gamma = 0$, (d) *Delay Factor* $\gamma = 60$

Figure: The efficiency ratios of our mechanism in various cases.

# Spectrum Utilization



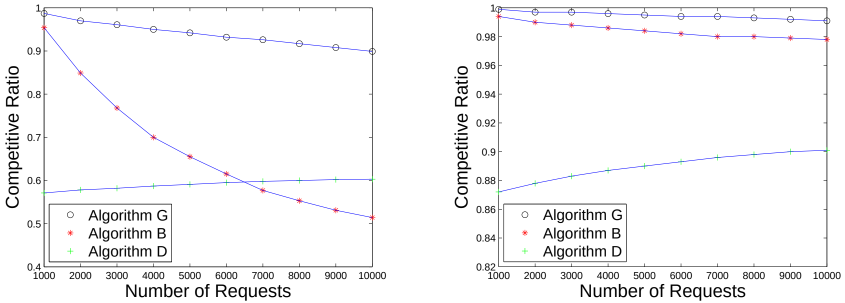

(a) *Delay Factor* $\gamma = 0$,




(b) *Delay Factor* $\gamma = 20$

Figure: The spectrum utilization ratios of method $\mathcal{G}$ in various cases.

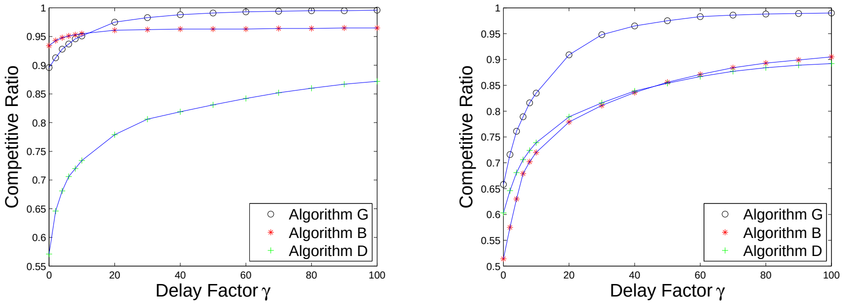
# Compared with other methods




(a) *Delay Factor* $\gamma = 0$, (b) *Delay Factor* $\gamma = 20$,

Figure: Compare algorithm $\mathcal{G}$ with two simple greedy algorithms.

# Compared with other methods




(c) 1,000 requests, (d) 10,000 requests

Figure: Compare algorithm $\mathcal{G}$ with two simple greedy algorithms.

# Summary

In this talk, we studied online spectrum allocation for wireless networks.

1. Instaneous Requests
   1. Designed efficient methods for $\beta < 1$, $\beta = 1$ and $\beta > 1$
   2. Present upper bounds for each of these cases.
   3. General penality function and conflict among nodes
   4. Designed truthful auction mechanisms (only manipulate bid *b*)

# Summary

In this talk, we studied online spectrum allocation for wireless networks.

1. Instaneous Requests
2. Requests in advance
   1. Designed efficient methods for $\beta < 1$, $\beta = 1$ and $\beta > 1$
   2. Find that $\alpha$ not affects performance ($\alpha > \gamma$)
   3. Present upper bounds for each of these cases.
   4. General penality function and conflict among nodes
   5. Designed auction mechanisms (only manipulate bid $b$), when each users will bid $b_i' \geq b_i$

# Future Work

1. Design allocation methods when we know probability distributions of requests (bid value, arrival time, and duration)
2. Design truthful online mechanisms (worst case and expected truthful)
3. Multiple channels for allocation (partial results done)
4. More general penality functions (partial results done)


**Thanks for your attention.**

**Questions and Comments?**