Online Scheduling for Delayed Mobile Offloading

I-Hong Hou

A Few Words On Performance

- Traditionally, “performance” is defined by asking the question:
- “Given a fixed amount of resource, what is the best service that can be provided?”
- This is a “best-effort” approach

A Few Words On Performance

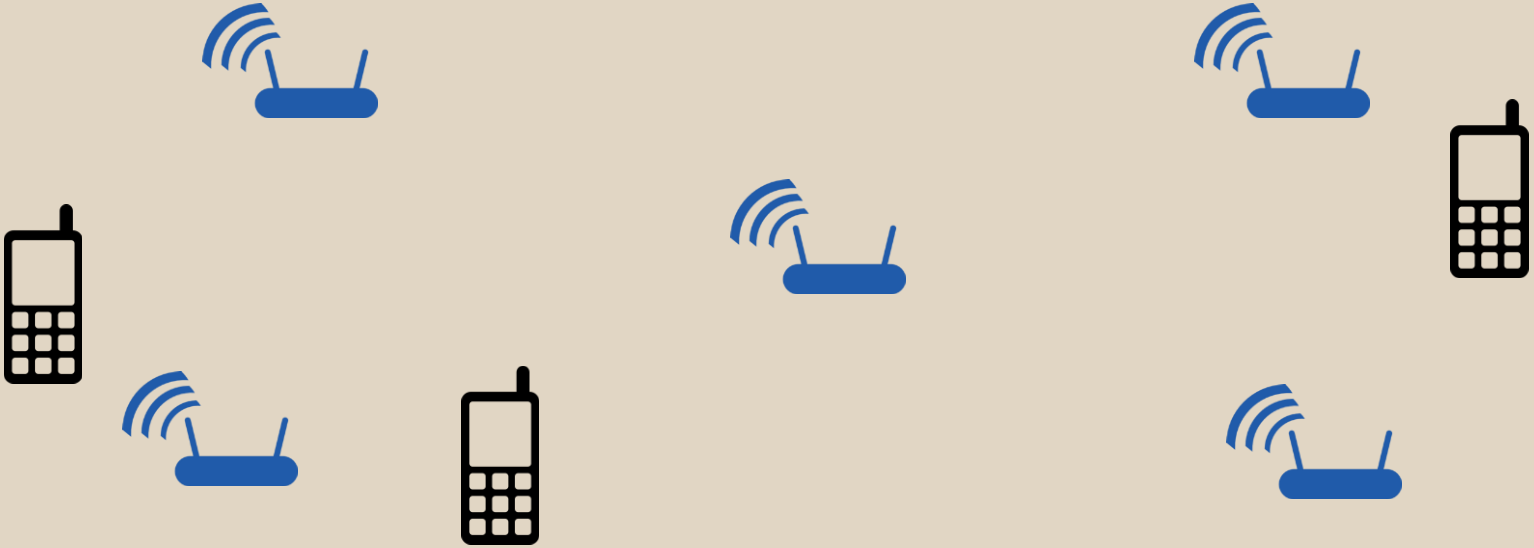
- Traditionally, “performance” is defined by asking the question:
- “Given a fixed amount of resource, what is the best service that can be provided?”
- This is a “best-effort” approach
- In this talk, we will ask a different question:
- “How much resource is needed to provide a fixed degree of service?”
- We will show that the second question reveals some surprising results

WiFi Offloading

- Wireless traffic is increasing at a fast pace
- Due to the limited bandwidth, cellular networks cannot keep up with the increase in demand
- An important approach to dramatically increase wireless capacity is to bring the transmitter and receiver closer, so as to reduce interference and improve spectrum efficiency
- WiFi offloading: Offload wireless traffic to the short-distance and low-power WiFi

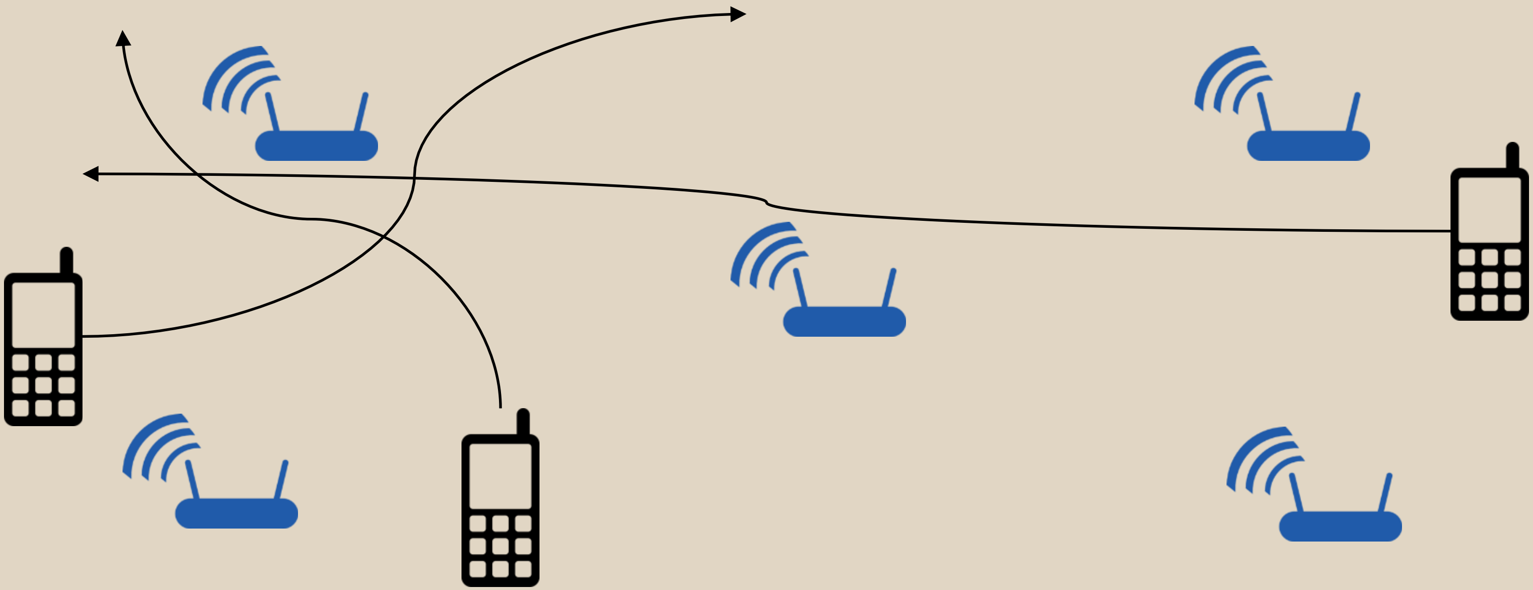
Problem Overview

- A system with M WiFi APs and I mobile users
- Users enter the system at different times
- Upon entrance, user i specifies the amount to data it needs to obtain, C_i, and a deadline



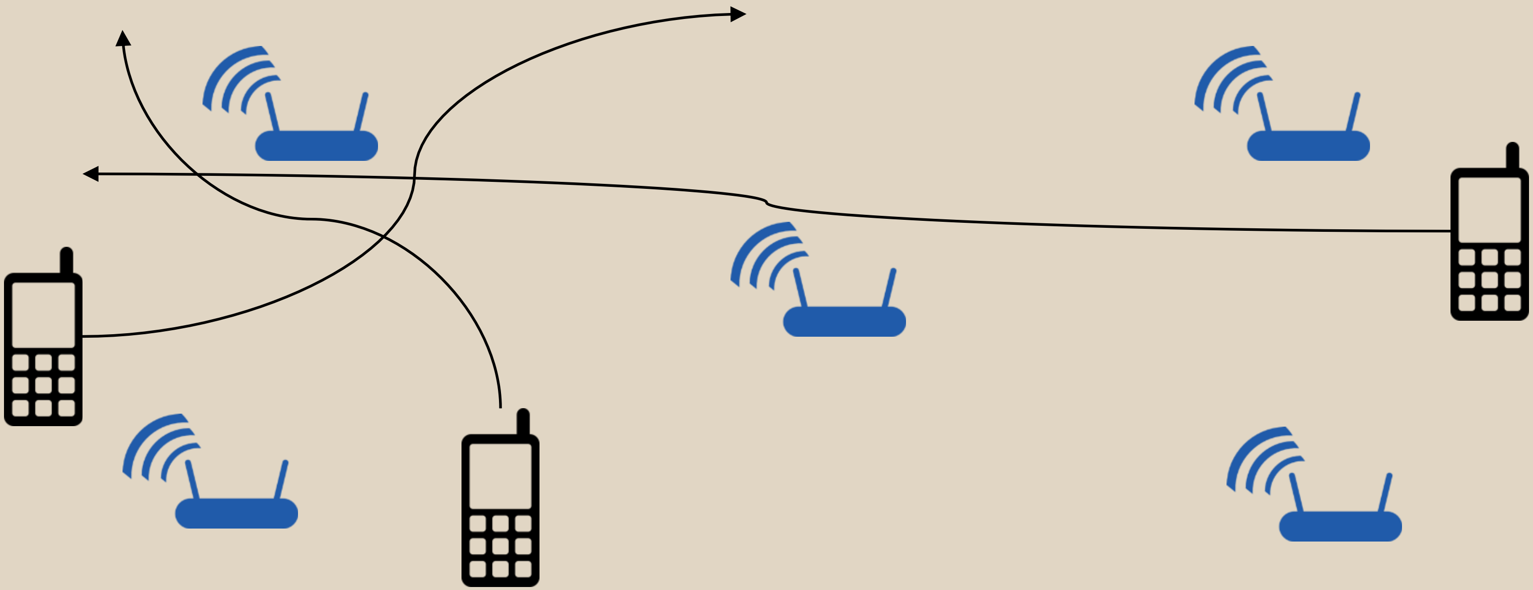
Problem Overview

- Users move around the system, and obtains data from WiFi opportunistically
- At the time of the deadlines, users download the rest data from 4G



Problem Overview

- Users are mobile, and we use K_{imt} to denote the channel rate between AP m and user i at time t
- AP allocates its capacity among connected users
- Goal: Maximize the amount of offloaded data



Given a fixed amount of resource, what is the best service that can be provided?

Linear Programming Formulation

- Let X_{imt} be the portion of bandwidth that AP m allocates to user i at time t
- $Max \sum_{imt} X_{imt} K_{imt}$ (maximize offloaded data)
- $s.t. \sum_{mt} X_{imt} K_{imt} \leq C_i, \forall i$ (user i only needs C_i)
- $\sum_i X_{imt} \leq 1, \forall m, t$ (capacity constraint)
- $X_{imt} \geq 0, \forall i, m, t$

- Challenge: We cannot know K_{imt} in advance
- We need to use online scheduling policies

Competitive Ratio

- Let Γ_{opt} be the amount of offloaded data using an optimal offline policy
  - The optimal offline policy knows all K_{imt} in advance
- Let Γ_{η} be the amount of offloaded data using policy η

- The competitive ratio is defined as the largest $\frac{\Gamma_{opt}}{\Gamma_{\eta}}$ over all possible C_i and K_{imt}

On/Off Channel

- On/Off channel: K_{imt} is either 0 or 1
- Work-conserving policy: AP schedules one arbitrary connected user that have yet to receive all data, as long as there is one
- Theorem: For On/Off channels, any work-conserving policy is 2-competitive
- (If the offline policy delivers all packets, any work-conserving policy delivers at least 50% of the packets)

An Online Policy for General Channels

- Each user i has a variable Z_i, which is initially 0
  - Z_i represents user i's progress
- In each time t, AP m schedules the client with the largest $K_{imt}(1 - Z_i)$
- If user i is served by AP m at time t, update Z_i by
  $Z_i \leftarrow Z_i \left(1 + \frac{K_{imt}}{C_i}\right) + \frac{K_{imt}}{(d-1)C_i}$,
  where $d = \left(1 + \frac{1}{C_{min}}\right)^{C_{min}}$

- Theorem: This policy is $\frac{e}{e-1}$-competitive

Some Observations

- Our online policy behaves pretty well, guaranteeing to offload at least 63% of data

Some Observations

- Our online policy behaves pretty well, guaranteeing to offload at least 63% of data
- Actually, any dumb policy works pretty well, guaranteeing to offload at least 50% of data
- It may not worth the effort to implement a better policy

- Our policy guarantees to offload at least 63% of data
- However, we need to ask…
- Where have all the undelivered packets gone?
- Gone to cellular, everyone
- Cellular networks are congested, which is the very reason of using WiFi offloading

How much resource is needed to offload, say, 95% of data?

Rethinking Competitive Ratio

- Assume that we have a hard requirement on the amount of data offloaded through WiFi
- We can increase the capacity of WiFi to meet the requirement
- When the capacity is increased by R:
- $Max \sum X_{imt} K_{imt}$
- $s.t. \sum_{mt} X_{imt} K_{imt} \le C_i, \forall i$
- $\sum_i X_{imt} \le R, \forall m, t$
- $X_{imt} \ge 0, \forall i, m, t$

New Definition

- Let Γ_{opt} be the amount of offloaded data using an optimal offline policy with unit capacity
  - The optimal offline policy knows all K_{imt} in advance
- Let $\Gamma_{\eta}(R)$ be the amount of offloaded data using policy η when the capacity is increased by R

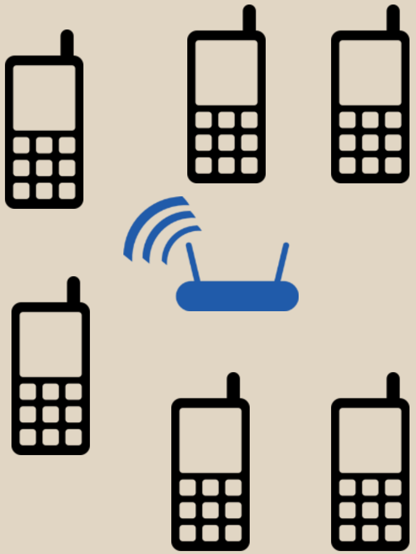
- Policy η is (R, β)-competitive if $\frac{\Gamma_{opt}}{\Gamma_{\eta}(R)} \leq \beta$ over all possible C_i and K_{imt}

(R, 100%)-Competitive

- Is it possible to offload all packets?
  - Achieves (R,100%)-competitive, for some R
- No!

(R, 100%)-Competitive

- Is it possible to offload all packets?
  - Achieves (R,100%)-competitive, for some R
- No!
- Ex. $2R$ users enter the system at the same time
- Each user requires the same amount of data
- One user leaves early, while other users stay forever
- It is not possible to offload all packets without knowing who leaves early



Performance Upper Bound

- Theorem:

  No policy is better than $(R, 1 + \frac{1}{2R})$-competitive, for all R

An Online Policy

- Each user i has a variable Z_i, which is initially 0
  - Z_i represents user i's progress
- In each time t, AP m schedules the client with the largest $K_{imt}(1 - Z_i)$
- If user i is served by AP m at time t, update Z_i by $Z_i \leftarrow Z_i\left(1 + \frac{K_{imt}}{C_i}\right) + \frac{K_{imt}}{(d-1)C_i}$,

  where $d = \left(1 + \frac{1}{C_{min}}\right)^{C_{min}/R}$

Performance of the Online Policy

- Theorem:

  This policy is $(R, \frac{e^{1/R}}{R[e^{1/R}-1]})$-competitive,

  which is approximately $(R, 1 + \frac{1}{2R})$-competitive

- Our policy almost achieves the performance upper bound

How about Work-Conserving Policy?

- We study the performance of round-robin policy for On/Off channel
- Round-robin: Evenly distribute capacity among all connected users that need more data

- Theorem:

  Round-robin is at best $(R, 1+\frac{1}{R})$-competitive

Some Comparison

- In order to offload 95% of data…

Some Comparison

- In order to offload 95% of data…
- The smallest R needs to have $1 + \frac{1}{2R} \leq \frac{1}{95\%} \Rightarrow R \geq 9.5$

Some Comparison

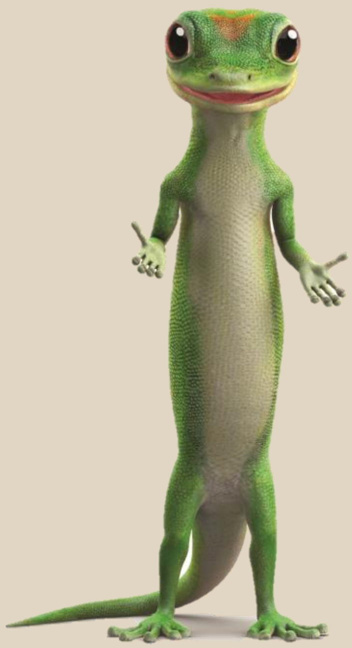
- In order to offload 95% of data…
- The smallest R needs to have $1+\frac{1}{2R}\leq\frac{1}{95\%}\Rightarrow R\geq 9.5$
- Our policy needs $R\geq 9.7$
  - $\frac{e^{1/R}}{R[e^{1/R}-1]}\leq\frac{1}{95\%}$, when $R\geq 9.7$

Some Comparison

- In order to offload 95% of data…
- The smallest R needs to have $1 + \frac{1}{2R} \leq \frac{1}{95\%} \Rightarrow R \geq 9.5$
- Our policy needs $R \geq 9.7$
  - $\frac{e^{1/R}}{R[e^{1/R}-1]} \leq \frac{1}{95\%}$, when $R \geq 9.7$
- Round-robin needs at least $R \geq 19$
- Compared to round-robin, our policy reduces capacity requirement by almost 50%

Summary

- We study WiFi offloading from both a fixed-resource view and a fixed-service view
- We develop an online policy that almost achieves the performance upper bound
- While our policy is only 13% better than round robin for a fixed amount of resource, it reduces the resource requirement by 50% when a strict performance guarantee is needed
- Implementing our policy is as effective as 2*2 MIMO



15% better efficiency can save you 50% or more on resource requirements