Capacity and Scheduling in Small-Cell HetNets

Stephen Hanly

Macquarie University
North Ryde, NSW 2109

Joint Work with Sem Borst, Chunshan Liu and Phil Whiting

Tuesday, January 13, 2015

Talk Summary

1. Small Cells and Research Challenges

Talk Summary

1. Small Cells and Research Challenges

2. Model

Talk Summary

1. Small Cells and Research Challenges
2. Model
3. Main Stability Results

Talk Summary

1. Small Cells and Research Challenges
2. Model
3. Main Stability Results
4. Discrete Linear Program

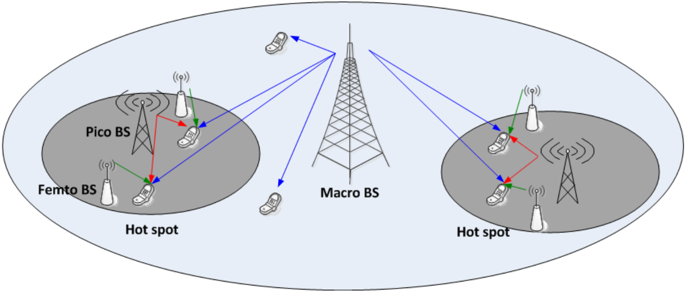
Talk Summary

1. Small Cells and Research Challenges
2. Model
3. Main Stability Results
4. Discrete Linear Program
5. Cell Association and Scheduling Algorithms

Talk Summary

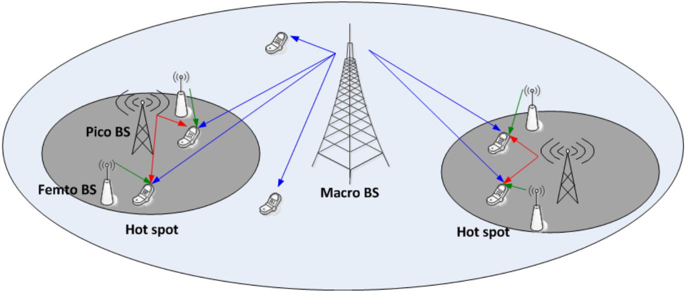
1. Small Cells and Research Challenges
2. Model
3. Main Stability Results
4. Discrete Linear Program
5. Cell Association and Scheduling Algorithms
6. Understanding the Converse

Small Cells and Data Offloading



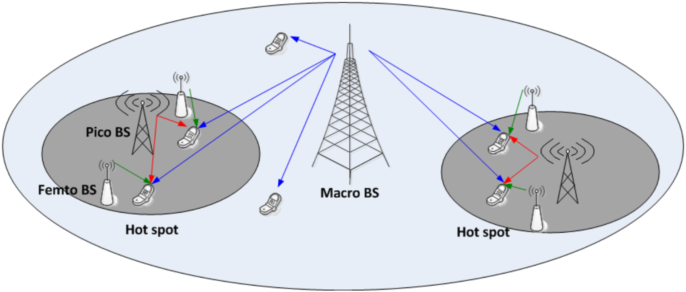
- Re-use spectrum: make cells smaller
- Offload traffic from macro-cells onto pico and femto-cells

Small Cells: Theoretical Challenges



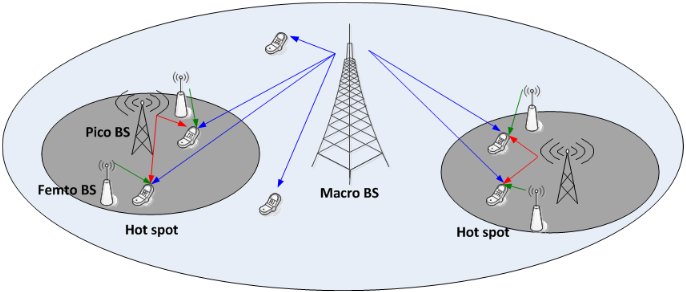
- What is the benefit? Base station densification gain?

Small Cells: Theoretical Challenges



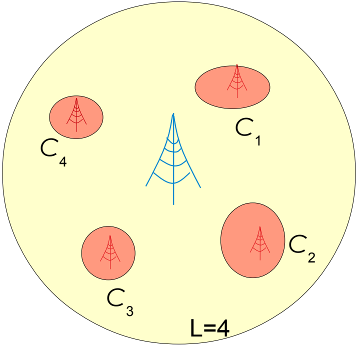
- What is the benefit? Base station densification gain?
- Characterizing capacity

Small Cells: Theoretical Challenges



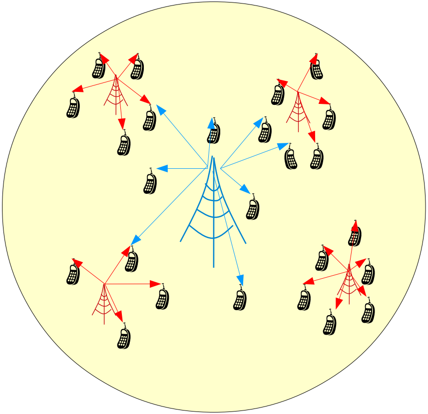
- What is the benefit? Base station densification gain?
- Characterizing capacity
- Optimizing resource allocation and cell association

System Model



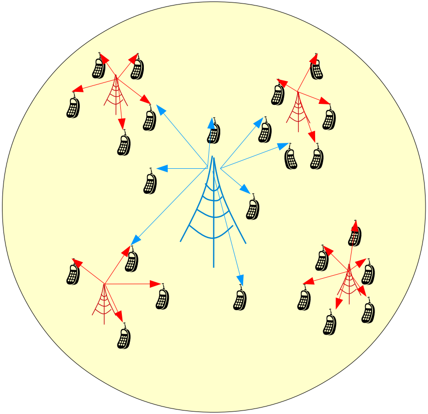
- One macro Base Station (BS)
- L pico BSs
- All users in coverage of macro BS
- $\mathcal{C}_\ell$ coverage area of pico BS ℓ
- Power levels are fixed
- Macro BS uses higher power than pico BSs

Time Sharing and Cell Association



- Time Share Spectrum
  - Macro Cell versus Pico Cells
  - Almost Blanking SubFrames

Time Sharing and Cell Association



- Time Share Spectrum
  - Macro Cell versus Pico Cells
  - Almost Blanking SubFrames
- Cell Range Expansion for Picos
  - Expand pico-cells to cover more mobiles
  - Contract pico-cells and send at Higher Rate

Research Problems

1. How to split the time between macro and picos?
2. How to decide the cell association?

Research Problems

1. How to split the time between macro and picos?
2. How to decide the cell association?

- For cell association, one way is via biasing: add a bias to the measured power level to encourage offloading.

Research Problems

1. How to split the time between macro and picos?
2. How to decide the cell association?

- For cell association, one way is via biasing: add a bias to the measured power level to encourage offloading.
- We will address both 1. and 2. in a joint approach

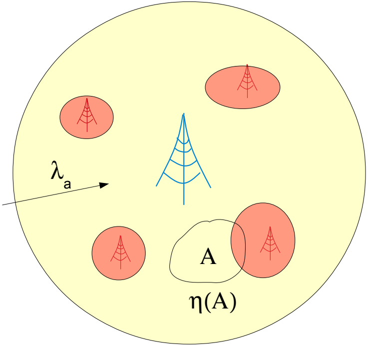
Research Problems

1. How to split the time between macro and picos?
2. How to decide the cell association?

- For cell association, one way is via biasing: add a bias to the measured power level to encourage offloading.
- We will address both 1. and 2. in a joint approach
- We will discover biasing based on rate ratios, not power levels

Talk Summary

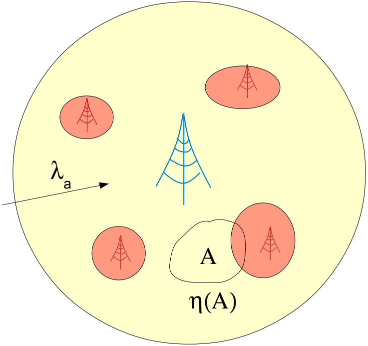
Arrivals in Space and Time



$\lambda(d\,\xi) = \lambda_a \;\; \eta(d\,\xi)$

- Files arrive at rate λ_a files/slot (Poisson)
- $\eta(\cdot)$ gives a probability measure on the macrocell area
- Spatial arrival intensity is $\lambda(d\xi) = \lambda_a \eta(d\xi)$

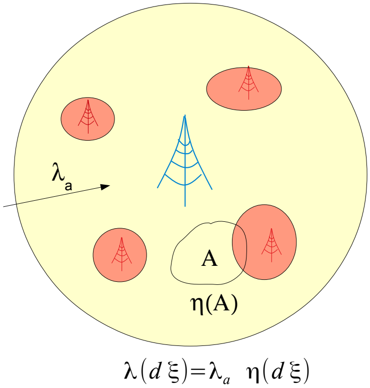
Arrivals in Space and Time



$\lambda(d\xi) = \lambda_a \ \eta(d\xi)$

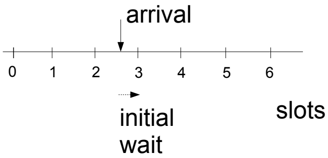
- Files arrive at rate λ_a files/slot (Poisson)
- $\eta(\cdot)$ gives a probability measure on the macrocell area
- Spatial arrival intensity is $\lambda(d\xi) = \lambda_a \eta(d\xi)$
- nth arrival has length D_n bits; $\mathbb{E}[D_n] = D$

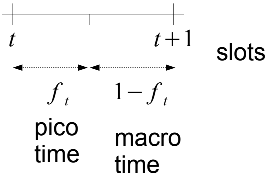
Arrivals in Space and Time



- Files arrive at rate λ_a files/slot (Poisson)
- $\eta(\cdot)$ gives a probability measure on the macrocell area
- Spatial arrival intensity is $\lambda(d\xi) = \lambda_a \eta(d\xi)$
- nth arrival has length D_n bits; $\mathbb{E}[D_n] = D$
- How large can λ_a be?

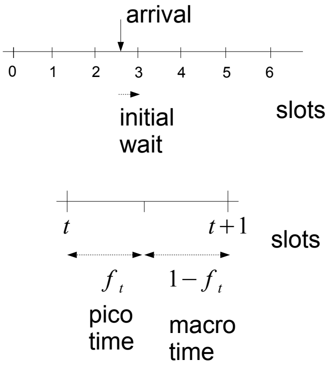
Pico versus Macro time





- Macro BS schedules files in macro-time
- All pico BSs can be scheduled simultaneously in pico-time

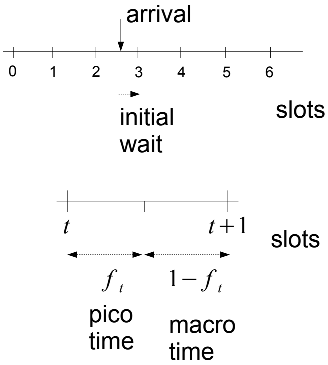
Pico versus Macro time



- Macro BS schedules files in macro-time
- All pico BSs can be scheduled simultaneously in pico-time

- A pico BS schedules files in its coverage area
- Not all files need be scheduled

Pico versus Macro time

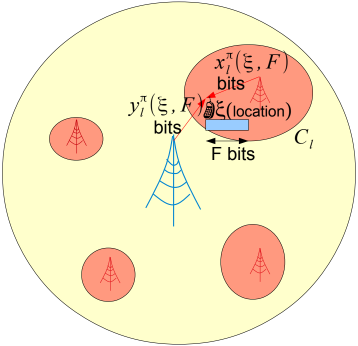


- Macro BS schedules files in macro-time
- All pico BSs can be scheduled simultaneously in pico-time

- A pico BS schedules files in its coverage area
- Not all files need be scheduled

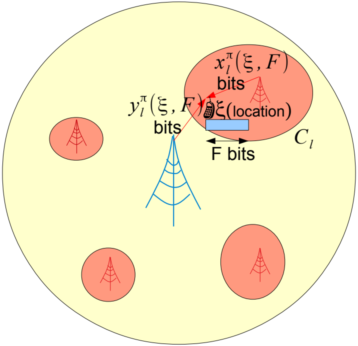
- We will consider only clearing schedules
- Clearing schedules include FCFS (one at a time) and PS (parallel processing)

Location based policies



- Schedules determined via only the location ξ and the size F (bits) of the file

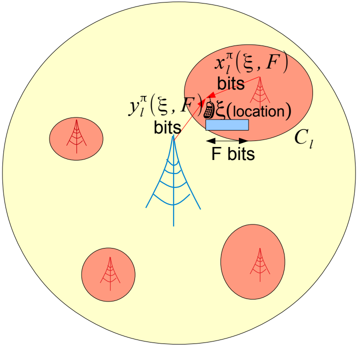
Location based policies



- Schedules determined via only the location ξ and the size F (bits) of the file

- If π is such a scheduler, $\xi \in \mathcal{C}_\ell$, and F is the file size then:
  - $x_\ell^\pi(\xi, F)$ bits are from pico BS ℓ
  - $y_\ell^\pi(\xi, F)$ are from macro BS

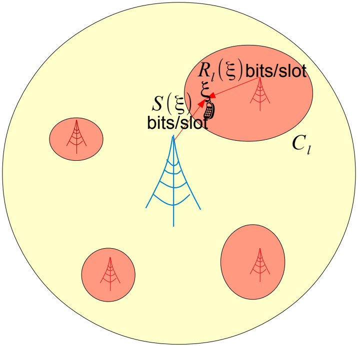
Location based policies



- Schedules determined via only the location ξ and the size F (bits) of the file
- If π is such a scheduler, $\xi \in \mathcal{C}_\ell$, and F is the file size then:
  - $x_\ell^\pi(\xi, F)$ bits are from pico BS ℓ
  - $y_\ell^\pi(\xi, F)$ are from macro BS
- If nth arrival is of size D_n bits and located at ξ_n then

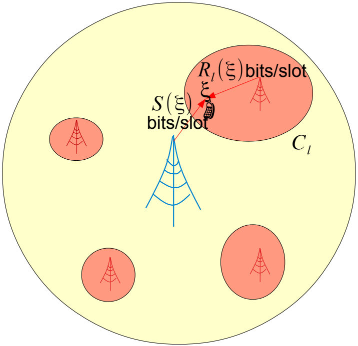
$$x_\ell^\pi(\xi_n, F_n) + y_\ell^\pi(\xi_n, F_n) = D_n$$

Data rates



- Assume no interference between picocells (will relax later)
- So there is no interference in the system!

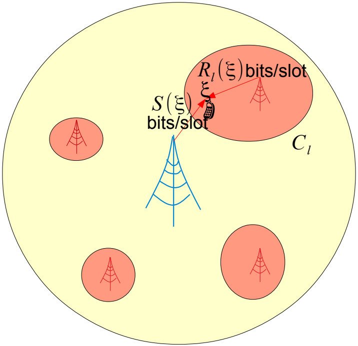
Data rates



- Assume no interference between picocells (will relax later)
- So there is no interference in the system!

- At any point ξ there is a macro-cell rate of $S(\xi)$ bits/slot
- At any point $\xi \in \mathcal{C}_\ell$ there is a pico-cell rate of $R_\ell(\xi)$ bits/slot

Data rates

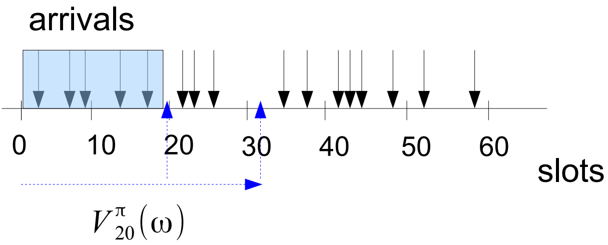


- Assume no interference between picocells (will relax later)
- So there is no interference in the system!

- At any point ξ there is a macro-cell rate of $S(\xi)$ bits/slot
- At any point $\xi \in \mathcal{C}_\ell$ there is a pico-cell rate of $R_\ell(\xi)$ bits/slot

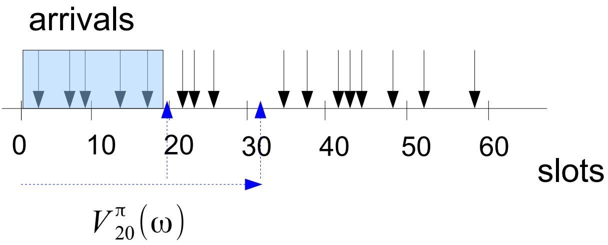
- A file can be served by macro and pico BSs in the same slot

Buildup of work



- Fix a location-based policy π and outcome $\omega \in \Omega$.
- Let $V_T^{\pi}(\omega)$ be time needed to clear all files that arrive in $[0, T]$

Buildup of work



- Fix a location-based policy π and outcome $\omega \in \Omega$.
- Let $V_T^{\pi}(\omega)$ be time needed to clear all files that arrive in $[0, T]$
- Clearly π is NOT stable if

$$\liminf_{T \uparrow \infty} \frac{V_T^{\pi}}{T} > 1$$

on an event of nonzero probability.

Talk Summary

Recall main parameters

$\lambda_a \eta(d\xi) = \lambda(d\xi)$, $\eta(d\xi)$ spatial intensity of arrivals

$R_\ell(\xi), S_\ell(\xi)$ Rates for pico and macro at location ξ

$x_\ell(\xi), y_\ell(\xi)$ bit assignments at location ξ

D mean download file size

Continuous Linear Program

Consider the following continuous LP:

$$\begin{aligned}
\min \quad & \tau = f + \sum_{\ell=1}^{L} \int \frac{y_\ell(\xi)}{S_\ell(\xi)} \lambda(d\xi) \\
\text{sub} \quad & \int \frac{x_\ell(\xi)}{R_\ell(\xi)} \lambda(d\xi) \le f \quad \forall \ell \\
& x_\ell(\xi) + y_\ell(\xi) \ge D,
\end{aligned}$$

where f represents pico-time.
Let τ^* be the optimal value of the program.

Sufficiency

Theorem (Hanly, Whiting)

Let τ^* be optimal solution to the LP. If $\tau^* < 1$, $\exists$ a clearing schedule π with ergodic properties.
Also define $S_n^\pi(\omega) :=$ sojourn time nth job, then π satisfies,

$$\mathbb{E}\left[S_n^\pi(\omega)\right] < \overline{S} < \infty \tag{1}$$

Converse

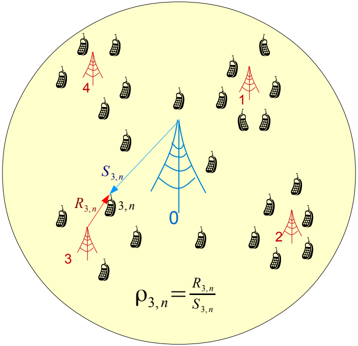
Theorem (Hanly, Whiting)

Let τ^* be the solution to the continuous LP. Suppose that $\tau^* > 1$ then there is a fixed constant $\eta > 0$, such that for any clearing schedule π

$$\liminf_{T\uparrow\infty} \frac{V_T^{\pi}(\omega)}{T} = 1 + \eta \quad \textit{almost surely.}$$

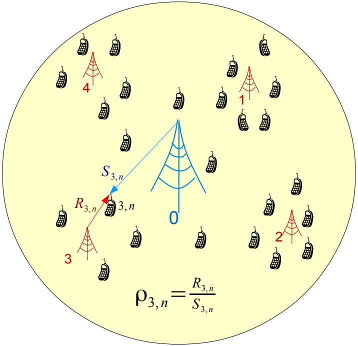
Talk Summary

Discrete Model

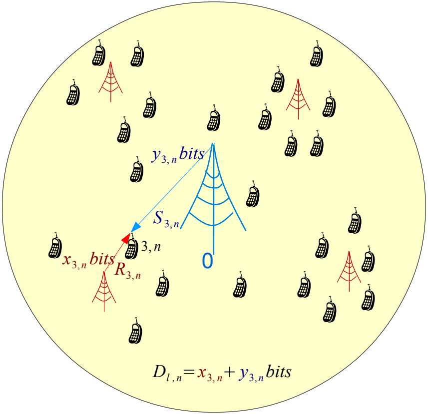


- Let instantaneous rate of *n*th user provided by pico BS be denoted by R_n
- Instantaneous rate of user provided by macro BS be denoted by S_n

Discrete Model

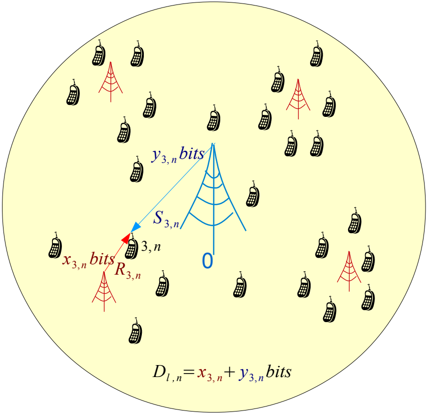


- Let instantaneous rate of nth user provided by pico BS be denoted by R_n
- Instantaneous rate of user provided by macro BS be denoted by S_n
- The rate ratio is defined by $\rho_n = \frac{R_n}{S_n}$



- Let D_n denote the amount of bits of data required by a user.
- D_n can be split into x_n bits of data from pico BS and y_n bits of data from the macro BS.

Problem Formulation



- Let D_n denote the amount of bits of data required by a user.
- D_n can be split into x_n bits of data from pico BS and y_n bits of data from the macro BS.
- It therefore requires $\frac{x_n}{R_n}$ secs from pico BS and $\frac{y_n}{S_n}$ secs from macro BS
- The problem is to minimize the total time to satisfy all the data demands in the network.

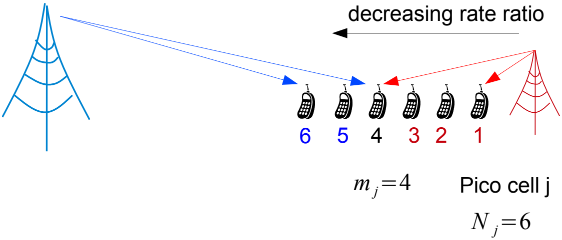
Problem Formulation: General

- The problem to be solved is the following linear program:

$$
\begin{aligned}
\min \quad & f + \sum_{l=0}^{L} \sum_{n=1}^{N_l} \frac{y_{l,n}}{S_{l,n}} \\
\text{sub} \quad & \sum_{n=1}^{N_l} \frac{x_{l,n}}{R_{l,n}} \leq f \quad \forall l \\
& x_{l,n} + y_{l,n} \geq D_{l,n} \quad \forall l, \ \forall n = 1, 2, \dots N_l \\
& f \geq 0, x_{l,n} \geq 0, y_{l,n} \geq 0 \quad \forall l, \ \forall n = 1, 2, \dots N_l
\end{aligned}
$$

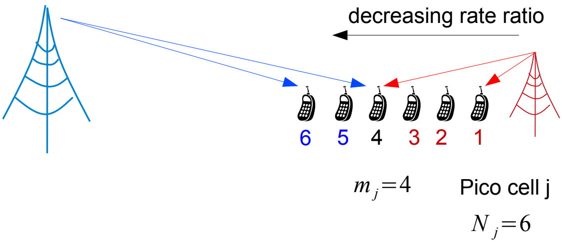
where f is the time allocated to the picocells.

A One Dimensional Formulation



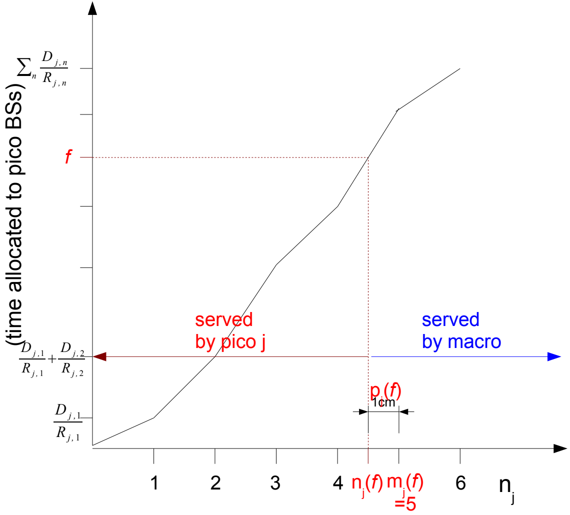
- Linear Programming theory tells us that the rate ratios $\rho := \frac{R}{S}$ are the key to the optimal cell association
- Order the users in pico cell in decreasing order of the rate ratio

A One Dimensional Formulation

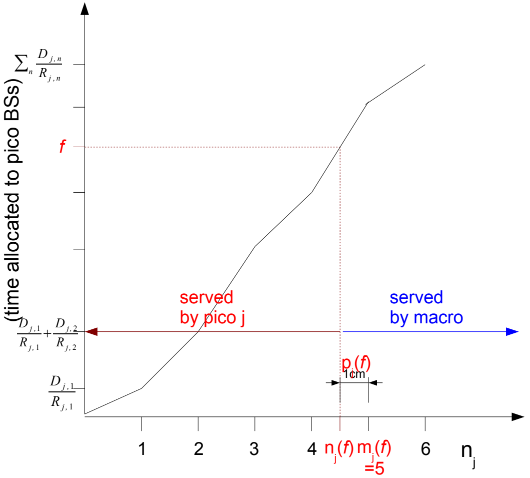


- Linear Programming theory tells us that the rate ratios $\rho := \frac{R}{S}$ are the key to the optimal cell association
- Order the users in pico cell in decreasing order of the rate ratio
- Then there will be a user m_j in pico cell j such that
  - users $1, 2, \ldots, m_j - 1$ will be 100 % served by the pico cell BS ($y_{j,n} = 0$ for these users)
  - users $m_j + 1, 2, \ldots, N_j$ will be 100 % served by the macro cell BS ($x_{j,n} = 0$ for these users)
  - user m_j may get its service from both base stations (pico and macro)

A One Dimensional Formulation



A one dimensional formulation



The macro-cell time required to service users users near pico cell j is

$$g_j(f) = p_j(f)\frac{D_{j,m_j(f)}}{S_{j,m_j(f)}} + \sum_{n=m_j(f)+1}^{N_j} \frac{D_{j,n}}{S_{j,n}}$$

A one dimensional formulation

The problem is therefore to minimize the following function of the scalar parameter f:

$$f + \sum_{l=1}^{L} g_l(f)$$

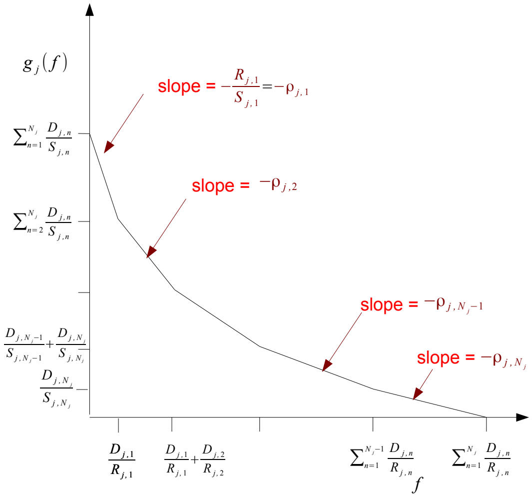
A one dimensional formulation

The problem is therefore to minimize the following function of the scalar parameter f:

$$f + \sum_{l=1}^{L} g_l(f)$$

The range for the optimization is $0 \leq f \leq \max_{l=1}^{L} \sum_{n=1}^{N_l} \frac{D_{l,n}}{S_{l,n}}$.

A one dimensional formulation



Edge rate condition

- Recall that the problem is to minimize the function $f + \sum_{l=1}^{L} g_l(f)$

Edge rate condition

- Recall that the problem is to minimize the function $f + \sum_{l=1}^{L} g_l(f)$
- The derivative at any *non-break-point* f is therefore $1 - \sum_{l=1}^{L} \rho_l(f)$.

Edge rate condition

- Recall that the problem is to minimize the function $f + \sum_{l=1}^{L} g_l(f)$
- The derivative at any *non-break-point* f is therefore $1 - \sum_{l=1}^{L} \rho_l(f)$.
- But the *optimum* will occur at one of the break points, where the derivative changes.

Edge rate condition

- Recall that the problem is to minimize the function $f + \sum_{l=1}^{L} g_l(f)$
- The derivative at any *non-break-point* f is therefore $1 - \sum_{l=1}^{L} \rho_l(f)$.
- But the *optimum* will occur at one of the break points, where the derivative changes.
- **edge rate condition:** The time allocation f to pico-cells is optimal if and only if

$$\sum_{l=1}^{L} \rho_{l,f-} \geq 1 \geq \sum_{l=1}^{L} \rho_{l,f+}$$

Edge rate condition

- Recall that the problem is to minimize the function $f + \sum_{l=1}^{L} g_l(f)$
- The derivative at any *non-break-point* f is therefore $1 - \sum_{l=1}^{L} \rho_l(f)$.
- But the *optimum* will occur at one of the break points, where the derivative changes.
- **edge rate condition:** The time allocation f to pico-cells is optimal if and only if

$$\sum_{l=1}^{L} \rho_{l,f-} \geq 1 \geq \sum_{l=1}^{L} \rho_{l,f+}$$

- We only need to check the $1 + \sum_{l=1}^{L} N_l$ break points, where the derivative changes, for the edge-rate condition.

Talk Summary

Recall theorem for existence of schedule

Theorem (Hanly, Whiting)

Let τ^* be optimal solution to the LP. If $\tau^* < 1$, $\exists$ a clearing schedule π with ergodic properties.

Recall theorem for existence of schedule

Theorem (Hanly, Whiting)

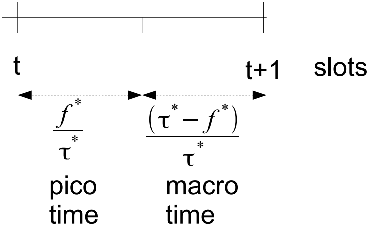
Let τ^* be optimal solution to the LP. If $\tau^* < 1$, $\exists$ a clearing schedule π with ergodic properties.

The optimal solution of the continuous LP is characterized by rate ratio thresholds $\rho_\ell^*, \quad \ell = 1, 2, \ldots, L$:

$$x_\ell^*(\xi) = \begin{cases} D & \rho_\ell(\xi) > \rho_\ell^* \\ 0 & \text{o.w.} \end{cases} \tag{2}$$

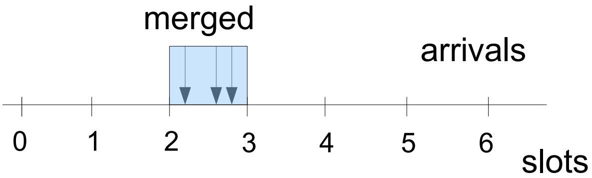
We will now show that these thresholds can be used to construct an optimal schedule.

Construction of a stabilizing schedule



- Suppose $\tau^* < 1$ and let f^* be the optimal value of f from continuous LP.
- Allocate $\frac{f^*}{\tau^*}$ of each slot to picos.
- Allocate $\frac{\tau^* - f^*}{\tau^*}$ of each slot to the macro.
- Assign each file to pico or macro based on rate-ratio threshold ρ_ℓ^*.

Construction of a stabilizing schedule



- All files that arrive during a slot are merged into one job
- Service time of each job can be computed (location-based policy)
- Each BS serves jobs in FCFS order
- D/G/1 queue at each base station.

Construction of a stabilizing schedule

- The workload arriving at the macro BS queue can be shown to be $\tau^* - f^*$ slots/slot
- The service rate of the macro BS is $\frac{\tau^* - f^*}{\tau^*}$ slots/slot
- So the utilization of the macro BS is τ^*

Construction of a stabilizing schedule

- The workload arriving at the macro BS queue can be shown to be $\tau^* - f^*$ slots/slot
- The service rate of the macro BS is $\frac{\tau^* - f^*}{\tau^*}$ slots/slot
- So the utilization of the macro BS is τ^*

- The workload arriving at a pico BS can be shown to be at most f^* slots/slot
- The service rate of each pico BS is $\frac{f^*}{\tau^*}$ slots/slot
- So the utilization of each pico BS is at most τ^*

Construction of a stabilizing schedule

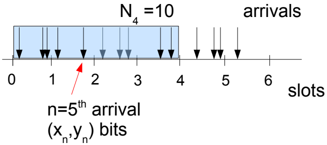
- The workload arriving at the macro BS queue can be shown to be $\tau^* - f^*$ slots/slot
- The service rate of the macro BS is $\frac{\tau^* - f^*}{\tau^*}$ slots/slot
- So the utilization of the macro BS is τ^*

- The workload arriving at a pico BS can be shown to be at most f^* slots/slot
- The service rate of each pico BS is $\frac{f^*}{\tau^*}$ slots/slot
- So the utilization of each pico BS is at most τ^*

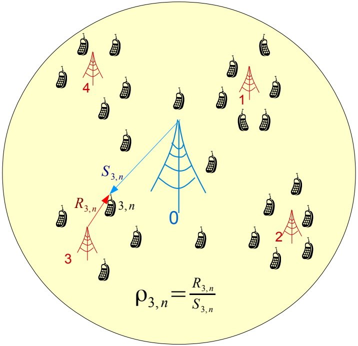
- If $\tau^* < 1$ then each D/G/1 queue is stable.

Talk Summary

Understanding the converse

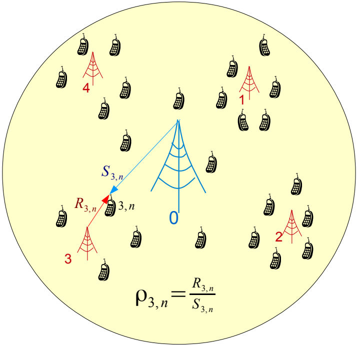


- Suppose $\tau^* > 1$, and let π be a clearing schedule.
- Let N_T be the number of arrivals during $[0, T]$.
- Let x_n be number of pico-cell bits for the nth arrival
- Let y_n be number of macro-cell bits for the nth arrival



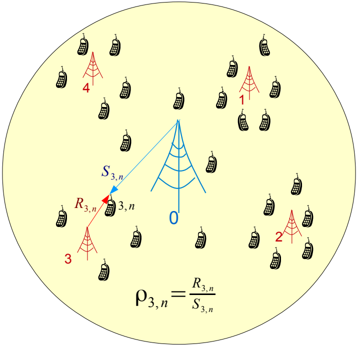
- Imagine all the N_T arrivals being present at time zero

Understanding the converse



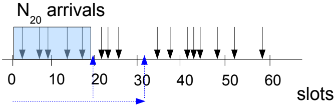
- Imagine all the N_T arrivals being present at time zero
- Let V_T^{LP} be the minimum time in the corresponding discrete LP

Understanding the converse



- Imagine all the N_T arrivals being present at time zero
- Let V_T^{LP} be the minimum time in the corresponding discrete LP
- We can analyze this and show that $\liminf_{T\uparrow\infty} \frac{V_T^{LP}}{T} > 1 + \eta$

Understanding the converse



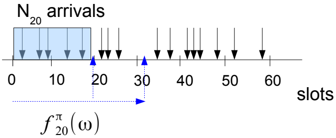
$f^{\pi}_{20}(\omega)$

= time when a pico is sending to at least one of N_{20} arrivals

Discrete Linear Program

$$\begin{aligned} min \quad & f + \sum_l \sum_n \frac{y_{l,n}}{S_{l,n}} \\ sub. \quad & \sum_n \frac{x_{l,n}}{R_{l,n}} \leq f \quad \forall l \\ & x_{l,n} + y_{l,n} \geq D_{l,n} \quad \forall l,n \end{aligned}$$

- Now consider the actual system under policy π
- Let f^{π}_{T} be the total time in which at least one of the N_T files is getting pico-service



= time when a pico is sending to at least one of N_{20} arrivals

Discrete Linear Program

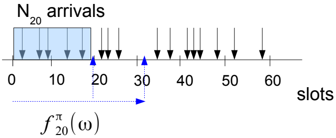
$$\begin{aligned} \min \quad & f + \sum_{l} \sum_{n} \frac{y_{l,n}}{S_{l,n}} \\ \text{sub.} \quad & \sum_{n} \frac{x_{l,n}}{R_{l,n}} \leq f \quad \forall l \\ & x_{l,n} + y_{l,n} \geq D_{l,n} \quad \forall l, n \end{aligned}$$

- Now consider the actual system under policy π
- Let f^{π}_{T} be the total time in which at least one of the N_T files is getting pico-service
- Then

$$V^{\pi}_{T} = f^{\pi}_{T} + \sum_{n=1}^{N_T} \frac{y_n}{S_n},$$

$$\sum_{n \in \mathcal{C}_\ell} \frac{x_n}{R_n} \leq f^{\pi}_{T}$$

Understanding the converse



= time when a pico is sending to at least one of N_{20} arrivals

Discrete Linear Program

$$\begin{aligned} \min \quad & f + \sum_{l} \sum_{n} \frac{y_{l,n}}{S_{l,n}} \\ \text{sub.} \quad & \sum_{n} \frac{x_{l,n}}{R_{l,n}} \leq f \quad \forall l \\ & x_{l,n} + y_{l,n} \geq D_{l,n} \quad \forall l,n \end{aligned}$$

- Now consider the actual system under policy π
- Let f_T^{π} be the total time in which at least one of the N_T files is getting pico-service
- Then

$$V_T^{\pi} = f_T^{\pi} + \sum_{n=1}^{N_T} \frac{y_n}{S_n},$$

$$\sum_{n \in \mathcal{C}_\ell} \frac{x_n}{R_n} \leq f_T^{\pi}$$

- So $(f_T^{\pi}, x_n, y_n, \ldots)$ is feasible for the earlier discrete LP

Understanding the converse

Hence

$$\liminf_{T \uparrow \infty} \frac{V_T^{\pi}}{T} \geq \frac{V_T^{LP}}{T} > 1 + \eta$$

which implies that π is unstable.

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved
- We can analyze Processor Sharing at each BS which gets much better delay performance

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved
- We can analyze Processor Sharing at each BS which gets much better delay performance
- Assumption that statistics are known can be relaxed

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved
- We can analyze Processor Sharing at each BS which gets much better delay performance
- Assumption that statistics are known can be relaxed
- The assumption that there is no pico-cell interference can be relaxed:
  - Assume fixed power levels when pico scheduled to be "on"

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved
- We can analyze Processor Sharing at each BS which gets much better delay performance
- Assumption that statistics are known can be relaxed
- The assumption that there is no pico-cell interference can be relaxed:
  - Assume fixed power levels when pico scheduled to be "on"
  - Many different modes: subsets of pico BSs that are simultaneously activated

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved
- We can analyze Processor Sharing at each BS which gets much better delay performance
- Assumption that statistics are known can be relaxed
- The assumption that there is no pico-cell interference can be relaxed:
  - Assume fixed power levels when pico scheduled to be "on"
  - Many different modes: subsets of pico BSs that are simultaneously activated
  - The pico rates at a location are mode-dependent

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved
- We can analyze Processor Sharing at each BS which gets much better delay performance
- Assumption that statistics are known can be relaxed
- The assumption that there is no pico-cell interference can be relaxed:
  - Assume fixed power levels when pico scheduled to be “on”
  - Many different modes: subsets of pico BSs that are simultaneously activated
  - The pico rates at a location are mode-dependent
  - Problem remains a continuous LP, but with a huge number of variables

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved
- We can analyze Processor Sharing at each BS which gets much better delay performance
- Assumption that statistics are known can be relaxed
- The assumption that there is no pico-cell interference can be relaxed:
  - Assume fixed power levels when pico scheduled to be "on"
  - Many different modes: subsets of pico BSs that are simultaneously activated
  - The pico rates at a location are mode-dependent
  - Problem remains a continuous LP, but with a huge number of variables
  - Good but suboptimal schemes can be found focusing on the most important modes

Extensions

- Given stabilizing schedule is NOT dynamic and can greatly be improved
- We can analyze Processor Sharing at each BS which gets much better delay performance
- Assumption that statistics are known can be relaxed
- The assumption that there is no pico-cell interference can be relaxed:
  - Assume fixed power levels when pico scheduled to be "on"
  - Many different modes: subsets of pico BSs that are simultaneously activated
  - The pico rates at a location are mode-dependent
  - Problem remains a continuous LP, but with a huge number of variables
  - Good but suboptimal schemes can be found focusing on the most important modes
- Extensions to multiple macro-cells

Applications

- Theory can be used as a basis for the search for adaptive algorithms
- Can be used as a cell planning tool since capacity is a function of base station locations

Concluding Remarks

- Formulated a notion of capacity for a dynamic HetNet consisting of one macrocell, multiple picocells

Concluding Remarks

- Formulated a notion of capacity for a dynamic HetNet consisting of one macrocell, multiple picocells
- Characterized capacity in terms of the solution of a deterministic, continuous Linear Program

Concluding Remarks

- Formulated a notion of capacity for a dynamic HetNet consisting of one macrocell, multiple picocells
- Characterized capacity in terms of the solution of a deterministic, continuous Linear Program
- Showed how ABS slot and cell association can be solved jointly

Concluding Remarks

- Formulated a notion of capacity for a dynamic HetNet consisting of one macrocell, multiple picocells
- Characterized capacity in terms of the solution of a deterministic, continuous Linear Program
- Showed how ABS slot and cell association can be solved jointly
- Showed how rate ratio thresholds provide the right way to bias pico-cells

Concluding Remarks

- Formulated a notion of capacity for a dynamic HetNet consisting of one macrocell, multiple picocells
- Characterized capacity in terms of the solution of a deterministic, continuous Linear Program
- Showed how ABS slot and cell association can be solved jointly
- Showed how rate ratio thresholds provide the right way to bias pico-cells
- Questions?