Pipes in the Sky

Resource Allocation in Wireless Networks

Siddhartha Banerjee

The University of Texas at Austin

April 20, 2009

Wireless Networking & Communications Group

THE UNIVERSITY OF TEXAS AT AUSTIN

Outline

Outline

How do 4 musicians play together?
such that you can hear what each one is playing...

The Problem

How do 4 musicians play together?
such that you can hear what each one is playing...

The Problem

- 4 musicians

How do 4 musicians play together?

such that you can hear what each one is playing...

The Problem

- 4 musicians
- Should play together

How do 4 musicians play together?

such that you can hear what each one is playing...

The Problem

- 4 musicians
- Should play together
- Should be heard independently

How do 4 musicians play together?

such that you can hear what each one is playing...

The Problem

- 4 musicians
- Should play together
- Should be heard independently

First Idea - Soloing

Make the musicians take turns.

How do 4 musicians play together?

such that you can hear what each one is playing...

Second Idea - Different Voices

Allot each musician a different 'voice'.

How do 4 musicians play together?

such that you can hear what each one is playing...

Second Idea - Different Voices

Allot each musician a different 'voice'.

How do 4 musicians play together?

such that you can hear what each one is playing...

Second Idea - Different Voices

Allot each musician a different 'voice'.

The Solution

- Different Voices + Soloing

How do 4 musicians play together?

such that you can hear what each one is playing...

Second Idea - Different Voices

Allot each musician a different 'voice'.

The Solution

- Different Voices + Soloing
- = string quartet.

Fine, but what about wireless communications?

The Problem

- How do n people simultaneously communicate.

The Solution

Fine, but what about wireless communications?

The Problem

- How do n people simultaneously communicate.
- Use techniques of wired communications.

The Solution

Fine, but what about wireless communications?

The Problem

- How do n people simultaneously communicate.
- ~~Use techniques of wired communications.~~
- Broadcast, superposition

The Solution

Fine, but what about wireless communications?

The Problem

- How do *n* people simultaneously communicate.
- ~~Use techniques of wired communications.~~
- Broadcast, superposition
- We need pipes in the sky!

The Solution

Fine, but what about wireless communications?

The Problem

- How do *n* people simultaneously communicate.
- ~~Use techniques of wired communications.~~
- Broadcast, superposition
- We need pipes in the sky!

The Solution

- Wireless channel $\longrightarrow$ 'resources'.

Fine, but what about wireless communications?

The Problem

- How do *n* people simultaneously communicate.
- ~~Use techniques of wired communications.~~
- Broadcast, superposition
- We need pipes in the sky!

The Solution

- Wireless channel $\longrightarrow$ 'resources'.
- Resources $\longrightarrow$ 'orthogonal pipes'.

Fine, but what about wireless communications?

The Problem

- How do *n* people simultaneously communicate.
- ~~Use techniques of wired communications.~~
- Broadcast, superposition
- We need pipes in the sky!

The Solution

- Wireless channel → 'resources'.
- Resources → 'orthogonal pipes'.
- Intelligent resource allocation

What are these resources?

Primary Channel Resources

- Time

What are these resources?

Primary Channel Resources

- Time
- Frequency

What are these resources?

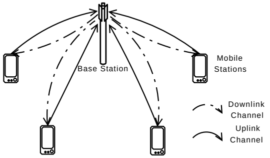
Primary Channel Resources

- Time
- Frequency
- (Space)

What are these resources?

Primary Channel Resources

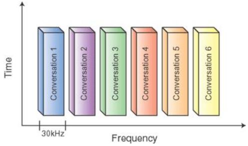
- Time
- Frequency
- (Space)

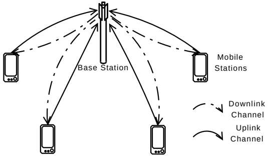


What are these resources?

Primary Channel Resources

- Time
- Frequency
- (Space)

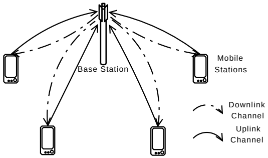


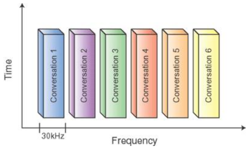


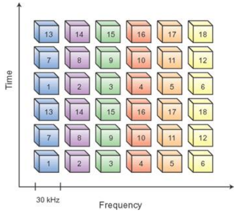
What are these resources?

Primary Channel Resources

- Time
- Frequency
- (Space)



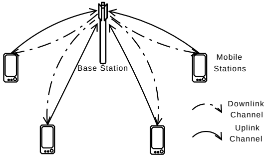


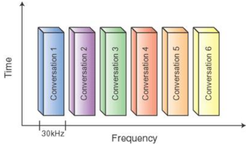


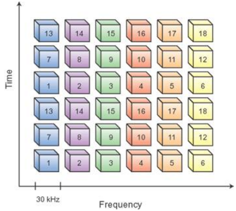
What are these resources?

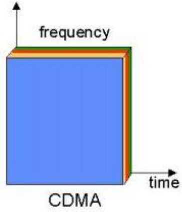
Primary Channel Resources

- Time
- Frequency
- (Space)









The stochastic nature of the wireless channel

Sources of Randomness

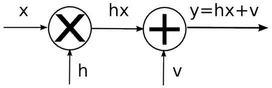
- Random channel coefficents

The stochastic nature of the wireless channel

Sources of Randomness

- Random channel coefficents
- Circuit noise

The stochastic nature of the wireless channel



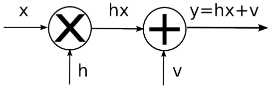
Sources of Randomness

- Random channel coefficents
- Circuit noise

The stochastic nature of the wireless channel

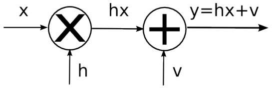
Sources of Randomness

- Random channel coefficents
- Circuit noise



$$y_i[n] = \sum_{j \in \mathcal{T}_n} h_{ji}[n] x_j[n] + v_i[n]$$

The stochastic nature of the wireless channel



$$y_i[n] = \sum_{j \in \mathcal{T}_n} h_{ji}[n] x_j[n] + v_i[n]$$

Sources of Randomness

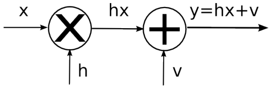
- Random channel coefficents
- Circuit noise

How do we deal with random channel coefficients

The stochastic nature of the wireless channel

Sources of Randomness

- Random channel coefficents
- Circuit noise



$$y_i[n] = \sum_{j \in \mathcal{T}_n} h_{ji}[n] x_j[n] + v_i[n]$$

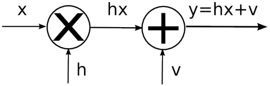
How do we deal with random channel coefficients

- Assumed to be fixed over a period of time called the coherence time

The stochastic nature of the wireless channel

Sources of Randomness

- Random channel coefficents
- Circuit noise



$$y_i[n] = \sum_{j \in \mathcal{T}_n} h_{ji}[n] x_j[n] + v_i[n]$$

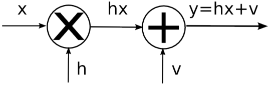
How do we deal with random channel coefficients

- Assumed to be fixed over a period of time called the coherence time
- Assumed to be i.i.d across these time slots

The stochastic nature of the wireless channel

Sources of Randomness

- Random channel coefficents
- Circuit noise



$$y_i[n] = \sum_{j \in \mathcal{T}_n} h_{ji}[n] x_j[n] + v_i[n]$$

How do we deal with random channel coefficients

- Assumed to be fixed over a period of time called the coherence time
- Assumed to be i.i.d across these time slots
- The channel coefficient in a time slot can be estimated.

The stochastic nature of the wireless channel

How do we deal with noise?

- Noise is noise limited

The stochastic nature of the wireless channel

How do we deal with noise?

- Noise is noise limited
- Idea: Increase transmit signal power

The stochastic nature of the wireless channel

How do we deal with noise?

- Noise is noise limited
- Idea: Increase transmit signal power
- Problem: Power is limited

The stochastic nature of the wireless channel

How do we deal with noise?

- Noise is noise limited
- Idea: Increase transmit signal power
- Problem: Power is limited

Resources

- Time, frequency

The stochastic nature of the wireless channel

How do we deal with noise?

- Noise is noise limited
- Idea: Increase transmit signal power
- Problem: Power is limited

Resources

- ~~Time, frequency~~
- Codes in a time slot

The stochastic nature of the wireless channel

How do we deal with noise?

- Noise is noise limited
- Idea: Increase transmit signal power
- Problem: Power is limited

Resources

- ~~Time, frequency~~
- Codes in a time slot
- Power

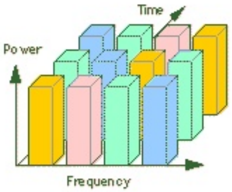
The stochastic nature of the wireless channel

How do we deal with noise?

- Noise is noise limited
- Idea: Increase transmit signal power
- Problem: Power is limited

Resources

- ~~Time, frequency~~
- Codes in a time slot
- Power



Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback
- Inhomogenous Flows

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback
- Inhomogenous Flows

What are the objectives?

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback
- Inhomogenous Flows

What are the objectives?

- Depend on system requirements.

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback
- Inhomogenous Flows

What are the objectives?

- Depend on system requirements.
- Long term averages.

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback
- Inhomogenous Flows

What are the objectives?

- Depend on system requirements.
- Long term averages.
- Examples:

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback
- Inhomogenous Flows

What are the objectives?

- Depend on system requirements.
- Long term averages.
- Examples:
  - Maximize system 'throughput'

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback
- Inhomogenous Flows

What are the objectives?

- Depend on system requirements.
- Long term averages.
- Examples:
  - Maximize system 'throughput'
  - Ensure fairness

Intelligent Wireless Resource Allocation

What factors motivate intelligent resource allocation?

- Packet switching
- Time discretization
- Channel coefficient estimate/feedback
- Inhomogenous Flows

What are the objectives?

- Depend on system requirements.
- Long term averages.
- Examples:
    - Maximize system 'throughput'
    - Ensure fairness
    - Stabilize queues

Problem Decomposition

Desired algorithm

- Online

Problem Decomposition

Desired algorithm

- Online
- Competitive

Problem Decomposition

Desired algorithm

- Online
- Competitive
- Low complexity

Problem Decomposition

Desired algorithm

- Online
- Competitive
- Low complexity

But observe...

Problem Decomposition

Desired algorithm

- Online
- Competitive
- Low complexity

But observe...

- Constraints are per time slot

Problem Decomposition

Desired algorithm

- Online
- Competitive
- Low complexity

But observe...

- Constraints are per time slot
- Channel is i.i.d across time slots

Problem Decomposition

Desired algorithm

- Online
- Competitive
- Low complexity

But observe...

- Constraints are per time slot
- Channel is i.i.d across time slots

Possible solution

Problem Decomposition

Desired algorithm

- Online
- Competitive
- Low complexity

But observe...

- Constraints are per time slot
- Channel is i.i.d across time slots

Possible solution

- Decompose to 'local' and 'global' problems.

Problem Decomposition

Desired algorithm

- Online
- Competitive
- Low complexity

But observe...

- Constraints are per time slot
- Channel is i.i.d across time slots

Possible solution

- Decompose to 'local' and 'global' problems.
- Local problem: Per time-slot optimal allocation.

Problem Decomposition

Desired algorithm

- Online
- Competitive
- Low complexity

But observe...

- Constraints are per time slot
- Channel is i.i.d across time slots

Possible solution

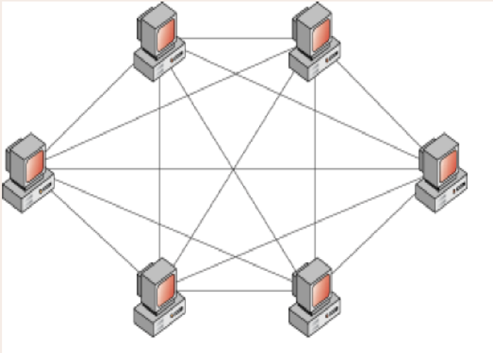
- Decompose to ‘local’ and ‘global’ problems.
- Local problem: Per time-slot optimal allocation.
- Global problem: Optimal choice of local objective function.

Outline

The Scheduling Problem for General Networks

How do we schedule on a network with-

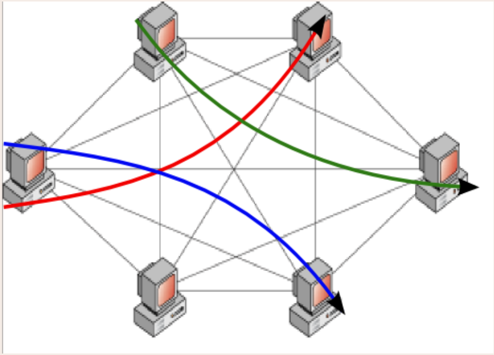
A General Communications System



The Scheduling Problem for General Networks

How do we schedule on a network with-

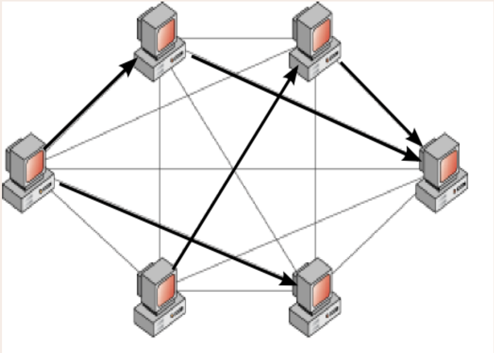
Multiple Input Flows



The Scheduling Problem for General Networks

How do we schedule on a network with-

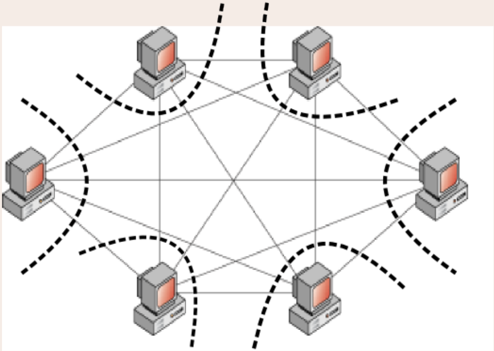
Random Communication Channels



The Scheduling Problem for General Networks

How do we schedule on a network with-

Constraints on Simultaneous Active Links



The Scheduling Problem for General Networks

How do we schedule on a network with-

- General topology
- Multiple flows
- Random channels
- Simultaneous transmission constraints

Aim of Scheduling Algorithm

- Given an input flow vector, stabilize queues

The Scheduling Problem for General Networks

How do we schedule on a network with-

- General topology
- Multiple flows
- Random channels
- Simultaneous transmission constraints

Aim of Scheduling Algorithm

- Given an input flow vector, stabilize queues
- Throughput Region: Set of all stabilizable input flows

The Scheduling Problem for General Networks

How do we schedule on a network with-

- General topology
- Multiple flows
- Random channels
- Simultaneous transmission constraints

Aim of Scheduling Algorithm

- Given an input flow vector, stabilize queues
- Throughput Region: Set of all stabilizable input flows
- Input rates unknown

The Back-Pressure Algorithm (BP)- An Overview

- Input: Current channel states ($h_{ij}[t]$) and queue states ($Q_i^d[t]$) in time slot t.

The Back-Pressure Algorithm (BP)- An Overview

- Input: Current channel states ($h_{ij}[t]$) and queue states ($Q_i^d[t]$) in time slot t.
- Output: Edges and flows to be activated.

The Back-Pressure Algorithm (BP)- An Overview

- Input: Current channel states ($h_{ij}[t]$) and queue states ($Q_i^d[t]$) in time slot t.
- Output: Edges and flows to be activated.
- Algorithm in 3 stages:

The Back-Pressure Algorithm (BP)- An Overview

- Input: Current channel states ($h_{ij}[t]$) and queue states ($Q_i^d[t]$) in time slot t.
- Output: Edges and flows to be activated.
- Algorithm in 3 stages:
  1. Weight w_i of edge $i =$'Back-pressure' across it.

The Back-Pressure Algorithm (BP)- An Overview

- Input: Current channel states ($h_{ij}[t]$) and queue states ($Q_i^d[t]$) in time slot t.
- Output: Edges and flows to be activated.
- Algorithm in 3 stages:
  1. Weight w_i of edge i ='Back-pressure' across it.
  2. Choose non-conflicting links to maximize $\sum w_i r_i$.

The Back-Pressure Algorithm (BP)- An Overview

- Input: Current channel states ($h_{ij}[t]$) and queue states ($Q_i^d[t]$) in time slot t.
- Output: Edges and flows to be activated.
- Algorithm in 3 stages:
  1. Weight w_i of edge $i =$'Back-pressure' across it.
  2. Choose non-conflicting links to maximize $\sum w_i r_i$.
  3. Activate chosen links

The Back-Pressure Algorithm (BP)- An Overview

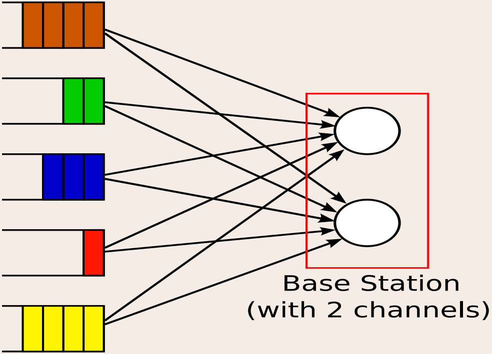
- Input: Current channel states ($h_{ij}[t]$) and queue states ($Q_i^d[t]$) in time slot t.
- Output: Edges and flows to be activated.
- Algorithm in 3 stages:
  1. Weight w_i of edge i =‘Back-pressure’ across it.
  2. Choose non-conflicting links to maximize $\sum w_i r_i$.
  3. Activate chosen links
- Per-time slot optimization → Long-term stability.

The Back-Pressure Algorithm (BP)- An Overview

- Input: Current channel states ($h_{ij}[t]$) and queue states ($Q_i^d[t]$) in time slot t.
- Output: Edges and flows to be activated.
- Algorithm in 3 stages:
  1. Weight w_i of edge $i =$'Back-pressure' across it.
  2. Choose non-conflicting links to maximize $\sum w_i r_i$.
  3. Activate chosen links
- Per-time slot optimization $\rightarrow$ Long-term stability.
- Input flow rates 'learnt' from queues.

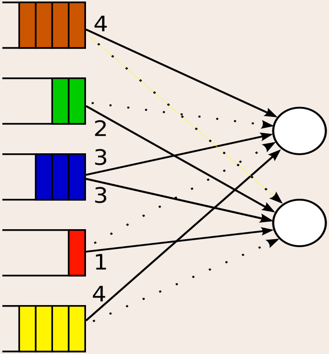
Example 1: Uplink Scheduling

The System



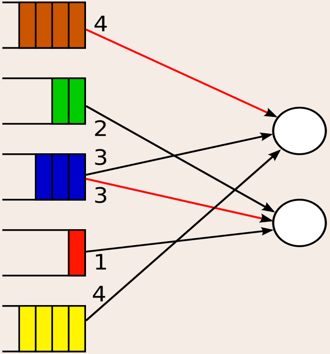
Example 1: Uplink Scheduling

BP Algorithm- Stage 1



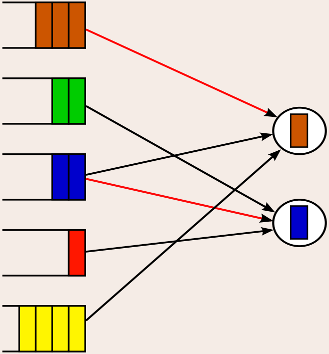
Example 1: Uplink Scheduling

BP Algorithm- Stage 2



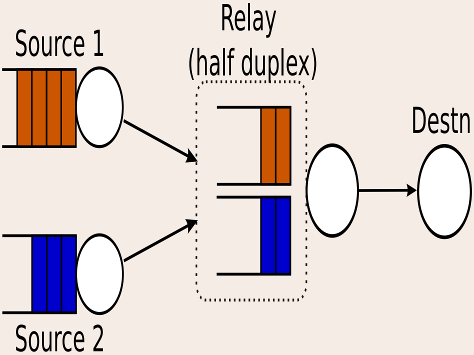
Example 1: Uplink Scheduling

BP Algorithm- Stage 3



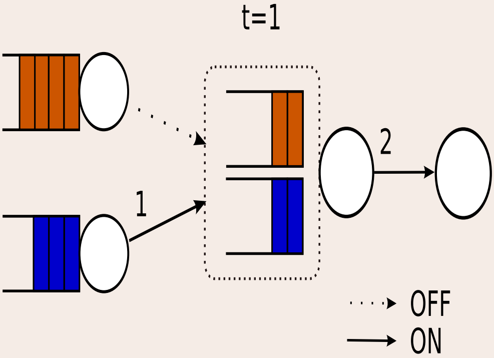
Example 2: Relay-aided Communications

The System



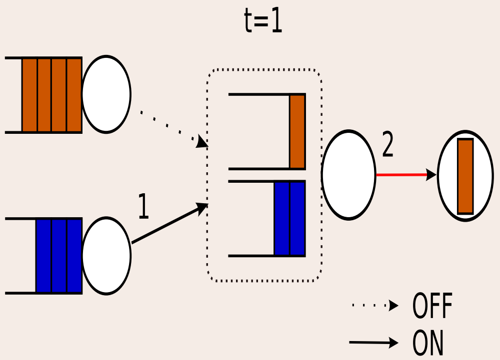
Example 2: Relay-aided Communications

First time slot



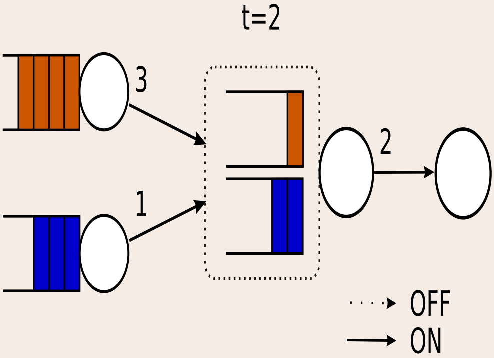
Example 2: Relay-aided Communications

First time slot



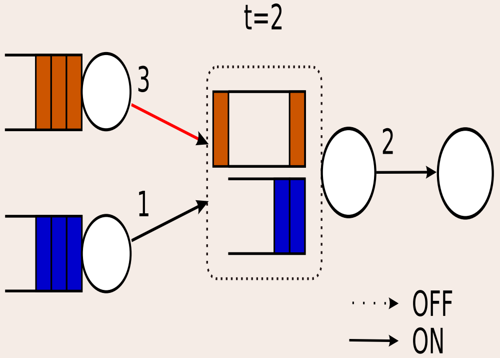
Example 2: Relay-aided Communications

Second time slot



Example 2: Relay-aided Communications

Second time slot



The Back-Pressure Algorithm (BP)- Formal Definition

The Algorithm

The Back-Pressure Algorithm (BP)- Formal Definition

The Algorithm

- Stage 1: Each edge e_{ij} calculates a weight $w_{ij}[t] = \max_{d \in V \setminus \{i,j\}}(Q_i^d[t] - Q_j^d[t])$ (the back-pressure across it).

The Back-Pressure Algorithm (BP)- Formal Definition

The Algorithm

- Stage 1: Each edge e_{ij} calculates a weight $w_{ij}[t] = \max_{d \in V \setminus \{i,j\}} (Q_i^d[t] - Q_j^d[t])$ (the back-pressure across it).
- Edge e_{ij} also stores $d_{ij}^* = \arg\max_{d \in V \setminus \{i,j\}} (Q_i^d[t] - Q_j^d[t])$ (the maximizing flow)

The Back-Pressure Algorithm (BP)- Formal Definition

The Algorithm

- Stage 1: Each edge e_{ij} calculates a weight $w_{ij}[t] = \max_{d \in V \setminus \{i,j\}} (Q_i^d[t] - Q_j^d[t])$ (the back-pressure across it).
- Edge e_{ij} also stores $d_{ij}^* = \arg\max_{d \in V \setminus \{i,j\}} (Q_i^d[t] - Q_j^d[t])$ (the maximizing flow)
- Stage 2: Choose a set of links $C^*[t] \in E$ such that-

$$C^*[t] = \arg \max_{C \subseteq V, C \in \mathcal{C}} \sum_{(i,j) \in C} w_{ij} h_{ij}$$

where $\mathcal{C}$ is the set of feasible simultaneous transmissions.

The Back-Pressure Algorithm (BP)- Formal Definition

The Algorithm

- Stage 1: Each edge e_{ij} calculates a weight $w_{ij}[t] = \max_{d \in V \setminus \{i,j\}}(Q_i^d[t] - Q_j^d[t])$ (the back-pressure across it).
- Edge e_{ij} also stores $d_{ij}^* = \arg\max_{d \in V \setminus \{i,j\}}(Q_i^d[t] - Q_j^d[t])$ (the maximizing flow)
- Stage 2: Choose a set of links $C^*[t] \in E$ such that-

$$C^*[t] = \arg \max_{C \subseteq V, C \in \mathcal{C}} \sum_{(i,j) \in C} w_{ij} h_{ij}$$

  where $\mathcal{C}$ is the set of feasible simultaneous transmissions.
- Stage 3: Activate all edges $e_{ij} \in C^*[t]$ and drain $Q_i^{d_{ij}^*}[t]$

Stability of Markov Chains

- System $\equiv$ vector of queues.

Stability of Markov Chains

- System $\equiv$ vector of queues.
- Queue vector is a Markov Chain.

Stability of Markov Chains

- System $\equiv$ vector of queues.
- Queue vector is a Markov Chain.
- Stability of scheduling rule $\equiv$ positive recurrence of the chain.

Stability of Markov Chains

- System $\equiv$ vector of queues.
- Queue vector is a Markov Chain.
- Stability of scheduling rule $\equiv$ positive recurrence of the chain.

Positive Recurrence of a Markov Chain

Stability of Markov Chains

- System $\equiv$ vector of queues.
- Queue vector is a Markov Chain.
- Stability of scheduling rule $\equiv$ positive recurrence of the chain.

Positive Recurrence of a Markov Chain

- A state i is recurrent if it is visited infinitely often, i.e., $\mathbb{P}(\text{eventual return to } i | X_0 = i) = 1$

Stability of Markov Chains

- System $\equiv$ vector of queues.
- Queue vector is a Markov Chain.
- Stability of scheduling rule $\equiv$ positive recurrence of the chain.

Positive Recurrence of a Markov Chain

- A state i is recurrent if it is visited infinitely often, i.e., $\mathbb{P}(\text{eventual return to } i | X_0 = i) = 1$
- A recurrent state i is positive recurrent if it has finite mean return time, i.e., $\mathbb{E}[\text{time taken to return to } i | X_0 = i] < \infty$

Lyapunov Function

Lyapunov Function

- $\mathbf{X} = \{X_n\}_{n \in \mathbf{N}}$: Irreducible DTMC, state space $\mathcal{X}$, transition probability matrix P

Lyapunov Function

- $\mathbf{X} = \{X_n\}_{n \in \mathbf{N}}$: Irreducible DTMC, state space $\mathcal{X}$, transition probability matrix P
- $\mathcal{L} : \mathcal{X} \to \mathcal{R}_+$: Candidate Lyapunov function

Lyapunov Function

- $\mathbf{X} = \{X_n\}_{n \in \mathbf{N}}$: Irreducible DTMC, state space $\mathcal{X}$, transition probability matrix P
- $\mathcal{L} : \mathcal{X} \to \mathcal{R}_+$: Candidate Lyapunov function
- $d_{\mathcal{L}}(x) = \mathbb{E}\left[\mathcal{L}(X_{n+1}) - \mathcal{L}(X_n) | X_n = x\right]$: Lyapunov drift

Lyapunov Function

- $\mathbf{X} = \{X_n\}_{n\in\mathbf{N}}$: Irreducible DTMC, state space $\mathcal{X}$, transition probability matrix P
- $\mathcal{L} : \mathcal{X} \to \mathcal{R}_+$: Candidate Lyapunov function
- $d_{\mathcal{L}}(x) = \mathbb{E}\left[\mathcal{L}(X_{n+1}) - \mathcal{L}(X_n) | X_n = x\right]$: Lyapunov drift

Foster's Theorem

Lyapunov Function

- $\mathbf{X} = \{X_n\}_{n\in\mathbf{N}}$: Irreducible DTMC, state space $\mathcal{X}$, transition probability matrix P
- $\mathcal{L} : \mathcal{X} \to \mathcal{R}_+$: Candidate Lyapunov function
- $d_{\mathcal{L}}(x) = \mathbb{E}\left[\mathcal{L}(X_{n+1}) - \mathcal{L}(X_n) | X_n = x\right]$: Lyapunov drift

Foster's Theorem

- $\exists$ finite set $A \subset X$ and constants $\varepsilon, \beta > 0$

Lyapunov Function

- $\mathbf{X} = \{X_n\}_{n \in \mathbf{N}}$: Irreducible DTMC, state space $\mathcal{X}$, transition probability matrix P
- $\mathcal{L} : \mathcal{X} \rightarrow \mathcal{R}_+$: Candidate Lyapunov function
- $d_{\mathcal{L}}(x) = \mathbb{E}\left[\mathcal{L}(X_{n+1}) - \mathcal{L}(X_n) | X_n = x\right]$: Lyapunov drift

Foster's Theorem

- $\exists$ finite set $A \subset X$ and constants $\varepsilon, \beta > 0$
- $d_{\mathcal{L}}(x) < \beta, \forall x \in A$

Lyapunov Function

- $\mathbf{X} = \{X_n\}_{n \in \mathbf{N}}$: Irreducible DTMC, state space $\mathcal{X}$, transition probability matrix P
- $\mathcal{L} : \mathcal{X} \rightarrow \mathcal{R}_+$: Candidate Lyapunov function
- $d_{\mathcal{L}}(x) = \mathbb{E}\left[\mathcal{L}(X_{n+1}) - \mathcal{L}(X_n) | X_n = x\right]$: Lyapunov drift

Foster's Theorem

- $\exists$ finite set $A \subset X$ and constants $\varepsilon, \beta > 0$
- $d_{\mathcal{L}}(x) < \beta, \forall x \in A$
- $d_{\mathcal{L}}(x) < -\varepsilon, \forall x \in \mathcal{X} \setminus A$

Lyapunov Function

- $\mathbf{X} = \{X_n\}_{n\in\mathbf{N}}$: Irreducible DTMC, state space $\mathcal{X}$, transition probability matrix P
- $\mathcal{L} : \mathcal{X} \to \mathcal{R}_+$: Candidate Lyapunov function
- $d_{\mathcal{L}}(x) = \mathbb{E}\left[\mathcal{L}(X_{n+1}) - \mathcal{L}(X_n) | X_n = x\right]$: Lyapunov drift

Foster's Theorem

- $\exists$ finite set $A \subset X$ and constants $\varepsilon, \beta > 0$
- $d_{\mathcal{L}}(x) < \beta, \forall x \in A$
- $d_{\mathcal{L}}(x) < -\varepsilon, \forall x \in \mathcal{X} \setminus A$
- Then $\mathbf{X}$ is positive recurrent.

Proof Overview for BP

- Lyapunov Function for BP: $\mathcal{L}(\mathcal{X}) = \sum_{i,d \in V, i \neq d} (Q_i^d)^2$

Proof Overview for BP

- Lyapunov Function for BP: $\mathcal{L}(\mathcal{X}) = \sum_{i,d \in V, i \neq d} (Q_i^d)^2$
- Intuition: Assume continuous flows in system (fluid limits).

Proof Overview for BP

- Lyapunov Function for BP: $\mathcal{L}(\mathcal{X}) = \sum_{i,d \in V, i \neq d} (Q_i^d)^2$
- Intuition: Assume continuous flows in system (fluid limits).
- Dynamics of queue:

$$\dot{q}_i^d(t) = \lambda_i^d(t) + \left[\mu_{\text{in}(i)}^d(t) - \mu_{\text{out}(i)}^d(t)\right]^+$$

Proof Overview for BP

- Lyapunov Function for BP: $\mathcal{L}(\mathcal{X}) = \sum_{i,d \in V, i \neq d} (Q_i^d)^2$
- Intuition: Assume continuous flows in system (fluid limits).
- Dynamics of queue:

$$\dot{q}_i^d(t) = \lambda_i^d(t) + \left[\mu_{\mathrm{in}(i)}^d(t) - \mu_{\mathrm{out}(i)}^d(t)\right]^+$$

  - $\lambda_i^d(t)$: Average external input flow for node $d$at node i

Proof Overview for BP

- Lyapunov Function for BP: $\mathcal{L}(\mathcal{X}) = \sum_{i,d \in V, i \neq d} (Q_i^d)^2$
- Intuition: Assume continuous flows in system (fluid limits).
- Dynamics of queue:

$$\dot{q}_i^d(t) = \lambda_i^d(t) + \left[\mu^d_{\mathrm{in}(i)}(t) - \mu^d_{\mathrm{out}(i)}(t)\right]^+$$

  - $\lambda_i^d(t)$: Average external input flow for node $d$at node i
  - $\mu^d_{\mathrm{in}(i)}(t)$:Total flow for node d in to node i

Proof Overview for BP

- Lyapunov Function for BP: $\mathcal{L}(\mathcal{X}) = \sum_{i,d \in V, i \neq d} (Q_i^d)^2$
- Intuition: Assume continuous flows in system (fluid limits).
- Dynamics of queue:

$$\dot{q}_i^d(t) = \lambda_i^d(t) + \left[\mu_{\text{in}(i)}^d(t) - \mu_{\text{out}(i)}^d(t)\right]^+$$

  - $\lambda_i^d(t)$: Average external input flow for node $d$at node i
  - $\mu_{\text{in}(i)}^d(t)$:Total flow for node d in to node i
  - $\mu_{\text{out}(i)}^d(t)$:Total flow for node d out of node i

Proof Overview for BP

- Lyapunov Function for BP: $\mathcal{L}(\mathcal{X}) = \sum_{i,d \in V, i \neq d} (Q_i^d)^2$
- Intuition: Assume continuous flows in system (fluid limits).
- Dynamics of queue:

$$\dot{q}_i^d(t) = \lambda_i^d(t) + \left[\mu_{\text{in}(i)}^d(t) - \mu_{\text{out}(i)}^d(t)\right]^+$$

  - $\lambda_i^d(t)$: Average external input flow for node $d$at node i
  - $\mu_{\text{in}(i)}^d(t)$:Total flow for node d in to node i
  - $\mu_{\text{out}(i)}^d(t)$:Total flow for node d out of node i
- Scheduling rule:

$$\arg\max_{C \in \mathcal{C}} \sum_{(i,j) \in C, d \in V} \max_{d \in V} \left[\mu_{ij}^d(t) \left(q_i^d(t) - q_j^d(t)\right)\right]$$

Proof Overview for BP

- $\mathcal{L}(X(t)) = \sum_{i,d \in V} \left(q_i^d(t)\right)^2$

Proof Overview for BP

- $\mathcal{L}(X(t)) = \sum_{i,d \in V} \left(q_i^d(t)\right)^2$
- $\dot{\mathcal{L}}(X(t)) = \sum_{i,d \in V} 2q_i^d(t)\dot{q}_i^d(t)$

Proof Overview for BP

- $\mathcal{L}(X(t)) = \sum_{i,d \in V} \left(q_i^d(t)\right)^2$
- $\dot{\mathcal{L}}(X(t)) = \sum_{i,d \in V} 2q_i^d(t)\dot{q}_i^d(t)$
- $\Rightarrow \dot{\mathcal{L}}(X(t)) =$
  $\sum_{i,d \in V} 2q_i^d(t) \left(\lambda_i^d(t) + \mu^d_{\mathrm{in}(i)}(t) - \mu^d_{\mathrm{out}(i)}(t)\right)$

Proof Overview for BP

- $\mathcal{L}(X(t)) = \sum_{i,d \in V} \left(q_i^d(t)\right)^2$
- $\dot{\mathcal{L}}(X(t)) = \sum_{i,d \in V} 2q_i^d(t)\dot{q}_i^d(t)$
- $\Rightarrow \dot{\mathcal{L}}(X(t)) =$
  $\sum_{i,d \in V} 2q_i^d(t)\left(\lambda_i^d(t) + \mu_{\text{in}(i)}^d(t) - \mu_{\text{out}(i)}^d(t)\right)$
- $\Rightarrow \dot{\mathcal{L}}(X(t)) =$
  $2\sum_{i,d \in V} q_i^d(t)\lambda_i^d(t) + \sum_{i,d \in V} q_i^d(t)\left(\mu_{\text{in}(i)}^d(t) - \mu_{\text{out}(i)}^d(t)\right)$

Proof Overview for BP

- $\mathcal{L}(X(t)) = \sum_{i,d\in V} \left(q_i^d(t)\right)^2$
- $\dot{\mathcal{L}}(X(t)) = \sum_{i,d\in V} 2q_i^d(t)\dot{q}_i^d(t)$
- $\Rightarrow \dot{\mathcal{L}}(X(t)) =$
  $\sum_{i,d\in V} 2q_i^d(t)\left(\lambda_i^d(t) + \mu^d_{\text{in}(i)}(t) - \mu^d_{\text{out}(i)}(t)\right)$
- $\Rightarrow \dot{\mathcal{L}}(X(t)) =$
  $2\sum_{i,d\in V} q_i^d(t)\lambda_i^d(t)+\sum_{i,d\in V} q_i^d(t)\left(\mu^d_{\text{in}(i)}(t) - \mu^d_{\text{out}(i)}(t)\right)$
- $\Rightarrow \dot{\mathcal{L}}(X(t)) =$
  $\sum_{i,d\in V} 2q_i^d(t)\lambda_i^d(t) - 2\sum_{(i,j)\in E,d\in V} \mu_{ij}^d(t)\left(q_i^d(t) - q_j^d(t)\right)$

Final Comments

Problem specific takeaways

Final Comments

Problem specific takeaways

1. Throughput region of a queuing system

Final Comments

Problem specific takeaways

1. Throughput region of a queuing system
2. The Back-Pressure algorithm

Final Comments

Problem specific takeaways

1. Throughput region of a queuing system
2. The Back-Pressure algorithm
3. Proof technique- Lyapunov function/Foster's Theorem

Final Comments

Problem specific takeaways

1. Throughput region of a queuing system
2. The Back-Pressure algorithm
3. Proof technique- Lyapunov function/Foster's Theorem

Big picture takeaways

Final Comments

Problem specific takeaways

1. Throughput region of a queuing system
2. The Back-Pressure algorithm
3. Proof technique- Lyapunov function/Foster's Theorem

Big picture takeaways

- Long term objective $\equiv$ Per-time slot optimization

Final Comments

Problem specific takeaways

1. Throughput region of a queuing system
2. The Back-Pressure algorithm
3. Proof technique- Lyapunov function/Foster's Theorem

Big picture takeaways

- Long term objective $\equiv$ Per-time slot optimization
- Local objective $\equiv$ Max weighted sum of rates

Final Comments

Problem specific takeaways

1. Throughput region of a queuing system
2. The Back-Pressure algorithm
3. Proof technique- Lyapunov function/Foster's Theorem

Big picture takeaways

- Long term objective $\equiv$ Per-time slot optimization
- Local objective $\equiv$ Max weighted sum of rates
- Need perfect CSI, feedback (of queue states).

Outline

Weighted Sum Rate Objective Functions

What is the rate of a flow?

Weighted Sum Rate Objective Functions

What is the rate of a flow?

- Amount of data transmitted in a time slot.

Weighted Sum Rate Objective Functions

What is the rate of a flow?

- Amount of data transmitted in a time slot.
- Shannon's formula: $r_i = B_i \; log_2 (1 + SINR_i)$

Weighted Sum Rate Objective Functions

What is the rate of a flow?

- Amount of data transmitted in a time slot.
- Shannon's formula: $r_i = B_i \ log_2 \left(1 + SINR_i\right)$

Weighted sum rate objectives

Weighted Sum Rate Objective Functions

What is the rate of a flow?

- Amount of data transmitted in a time slot.
- Shannon's formula: $r_i = B_i \; log_2 \left(1 + SINR_i\right)$

Weighted sum rate objectives

- General form:

$$V^*[n] = \max_{\mathbf{r}[n] \in \mathcal{R}} \sum_i w_i[n] r_i[n]$$

Weighted Sum Rate Objective Functions

What is the rate of a flow?

- Amount of data transmitted in a time slot.
- Shannon's formula: $r_i = B_i \; log_2 \left(1 + SINR_i\right)$

Weighted sum rate objectives

- General form:

$$V^*[n] = \max_{\mathbf{r}[n] \in \mathcal{R}} \sum_i w_i[n] r_i[n]$$

- w_i function of QOS metrics of user i (E.g. Throughput T_i, HOL packet delay D_i, queue length Q_i, etc.)

Resource Allocation Problem

Objective Function

$$V^* = \max_{\mathbf{r} \in \mathcal{R}(\mathbf{h})} \sum_i w_i r_i$$

Where rate is given by

$$r_i = n_i B \ log_2 \left(1 + \frac{p_i h_i}{n_i} \right)$$

Resource Allocation Problem

Objective Function

$$V^* = \max_{\mathbf{r} \in \mathcal{R}(\mathbf{h})} \sum_i w_i r_i$$

Where rate is given by

$$r_i = n_i B \ log_2 \left(1 + \frac{p_i h_i}{n_i}\right)$$

Constraints

Resource Allocation Problem

Objective Function

$$V^* = \max_{\mathbf{r} \in \mathcal{R}(\mathbf{h})} \sum_i w_i r_i$$

Where rate is given by

$$r_i = n_i B \; log_2 \left(1 + \frac{p_i h_i}{n_i}\right)$$

Constraints

- Non-negativity: $n_i \in \mathbb{Z}^+, p_i \in \mathbb{R}^+$

Resource Allocation Problem

Objective Function

$$V^* = \max_{\mathbf{r} \in \mathcal{R}(\mathbf{h})} \sum_i w_i r_i$$

Where rate is given by

$$r_i = n_i B \ log_2 \left(1 + \frac{p_i h_i}{n_i} \right)$$

Constraints

- Non-negativity: $n_i \in \mathbb{Z}^+, p_i \in \mathbb{R}^+$
- Maximum Codes Constraint: $\sum_i n_i \leq N$

Resource Allocation Problem

Objective Function

$$V^* = \max_{\mathbf{r} \in \mathcal{R}(\mathbf{h})} \sum_i w_i r_i$$

Where rate is given by

$$r_i = n_i B \ log_2 \left(1 + \frac{p_i h_i}{n_i} \right)$$

Constraints

- Non-negativity: $n_i \in \mathbb{Z}^+, p_i \in \mathbb{R}^+$
- Maximum Codes Constraint: $\sum_i n_i \leq N$
- Maximum Power Constraint: $\sum_i p_i \leq P$

Optimal Solution

- The problem is non-linear, mixed integer.

Optimal Solution

- The problem is non-linear, mixed integer.
- Concave objective function, convex feasible set (under given constraints).

Optimal Solution

- The problem is non-linear, mixed integer.
- Concave objective function, convex feasible set (under given constraints).
- There is no duality gap (by Slater's condition).

Optimal Solution

- The problem is non-linear, mixed integer.
- Concave objective function, convex feasible set (under given constraints).
- There is no duality gap (by Slater's condition).

Optimal Solution using Lagrange multipliers (overview)

Optimal Solution

- The problem is non-linear, mixed integer.
- Concave objective function, convex feasible set (under given constraints).
- There is no duality gap (by Slater's condition).

Optimal Solution using Lagrange multipliers (overview)

- Dual of problem: $\min_{(\lambda,\mu)\geq 0} \max_{(\mathbf{x},\mathbf{p})\in\mathcal{X}} L(\lambda,\mu,\mathbf{x},\mathbf{p})$, where:

$$L(\lambda,\mu,\mathbf{x},\mathbf{p}) := \sum_i w_i r_i + \lambda(P_i - \sum_i p_i) + \mu(N - \sum_i n_i)$$

Optimal Solution

- The problem is non-linear, mixed integer.
- Concave objective function, convex feasible set (under given constraints).
- There is no duality gap (by Slater's condition).

Optimal Solution using Lagrange multipliers (overview)

- Dual of problem: $\min_{(\lambda,\mu)\geq 0} \max_{(\mathbf{x},\mathbf{p})\in\mathcal{X}} L(\lambda,\mu,\mathbf{x},\mathbf{p})$, where:

$$L(\lambda,\mu,\mathbf{x},\mathbf{p}) := \sum_i w_i r_i + \lambda(P_i - \sum_i p_i) + \mu(N - \sum_i n_i)$$

- Optimize over p_i, then n_i, then μ_i in terms of λ_i.

Optimal Solution

- The problem is non-linear, mixed integer.
- Concave objective function, convex feasible set (under given constraints).
- There is no duality gap (by Slater's condition).

Optimal Solution using Lagrange multipliers (overview)

- Dual of problem: $\min_{(\lambda,\mu)\geq 0} \max_{(\mathbf{x},\mathbf{p})\in\mathcal{X}} L(\lambda, \mu, \mathbf{x}, \mathbf{p})$, where:

$$L(\lambda, \mu, \mathbf{x}, \mathbf{p}) := \sum_i w_i r_i + \lambda(P_i - \sum_i p_i) + \mu(N - \sum_i n_i)$$

- Optimize over p_i, then n_i, then μ_i in terms of λ_i.
- λ_i found using gradient descent.

Final Comments

Comments on the Local Problem

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.
- Suboptimal algorithms proposed.

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.
- Suboptimal algorithms proposed.
- Uplink scheduling is a simple extension.

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.
- Suboptimal algorithms proposed.
- Uplink scheduling is a simple extension.
- Has been extended to other multiple access schemes.

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.
- Suboptimal algorithms proposed.
- Uplink scheduling is a simple extension.
- Has been extended to other multiple access schemes.

Overall Comments

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.
- Suboptimal algorithms proposed.
- Uplink scheduling is a simple extension.
- Has been extended to other multiple access schemes.

Overall Comments

- Resource allocation=Local+Global problem

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.
- Suboptimal algorithms proposed.
- Uplink scheduling is a simple extension.
- Has been extended to other multiple access schemes.

Overall Comments

- Resource allocation=Local+Global problem
- Local Problem: Per-time slot max weighted sum rate

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.
- Suboptimal algorithms proposed.
- Uplink scheduling is a simple extension.
- Has been extended to other multiple access schemes.

Overall Comments

- Resource allocation=Local+Global problem
- Local Problem: Per-time slot max weighted sum rate
- Global Problem: Choosing the weights

Final Comments

Comments on the Local Problem

- General framework for per-time slot problem.
- Optimal algorithm is too slow for practical use.
- Suboptimal algorithms proposed.
- Uplink scheduling is a simple extension.
- Has been extended to other multiple access schemes.

Overall Comments

- Resource allocation=Local+Global problem
- Local Problem: Per-time slot max weighted sum rate
- Global Problem: Choosing the weights
- Feedback is essential, assumed perfect