# Lec18-mwf

Sunday, February 11, 2018 12:29 PM

Midterm 1:

- MSEE 239, 6:30-7:30pm Tuesday 3/10.
- Open-book, open-notes. However, you cannot bring other materials, such as homework, solutions, or papers.
- Cover topics from the beginning of the semester up to optimal cross-layer control (today's lecture).
- Sample exam on the web.

# Back-pressure - 15min

Friday, February 01, 2008 3:40 PM

- Optimal Joint Routing & Scheduling

Model:

- Link model same as that in the scheduling problem

$$\vec{r} = g(\vec{P}, K(t))$$

$$\vec{P} \in \Theta$$

$K(t)$ i.i.d over time.

- Multiple commodities $c = 1, 2, \cdots C$, each with a destination node $d(c)$

- A number of nodes may generate packets of commodity $c$.

  $\lambda_i^c$ = rate of new packets of commodity $c$ generated by node $i$

  - No routes are specified yet.

- In order for the arrival rate vector $[\lambda_i^c]$ to be supported, there must exist $[\overline{r}_{ij}^c]$ such that

  - $\overline{r}_{ij}^c$ is the rate that node $i$ forwards commodity $c$ to node $j$

  - $$\lambda_i^c + \sum_{j \neq i} \overline{r}_{ji} \leq \sum_{j \neq i} \overline{r}_{ij}$$

    for all node $i \neq d(c)$

  - $$\left[\sum_c \overline{r}_{ij}^c\right] \in \Lambda \triangleq \sum_k \lambda_k \, \text{Conv-Hull}\{g(\vec{P}, k) \mid \vec{P} \in \Theta\}$$

# Queue-length Based Policy

- Each node maintains multiple queues, one for each destination $c$
  $q_i^c$

  Let $r_{ij}^c$ be the amount of packets of commodity $c$ that are forwarded from node $i$ to node $j$.

$$q_i^c(t+1) = \begin{cases} \left[ q_i^c(t) + \sum_{j \neq i} r_{ji}^c + \lambda_i^c - \sum_{j \neq i} r_{ij}^c \right]^+ & \text{if } i \neq d(c) \\ = 0 & \text{if } i = d(c). \end{cases}$$

---

- We now see why working with the Lyapunov function also produces a joint routing & scheduling policy that is throughput-optimal

- Use the Lyapunov function

$$V(\vec{q}) = \frac{1}{2} \sum_{i,c} (q_i^c)^2$$

Note

$$\left(q_i^c(t+1)\right)^2 - \left(q_i^c(t)\right)^2$$

$$\leq 2 q_i^c(t) \left[ \sum_{j \neq i} r_{ji}^c + \lambda_i^c - \sum_{j \neq i} r_{ij}^c \right] + \text{constant}$$

$$\Rightarrow \quad V(\vec{q}(t+1)) - V(\vec{q}(t))$$

$$\leq \sum_{i,c} q_i^c(t) \left[ \sum_{j \neq i} r_{ji}^c + \lambda_i^c - \sum_{j \neq i} r_{ij}^c \right] + \text{constant}$$

$$= \sum_{i,c} q_i^c(t) \cdot \lambda_i^c - \underbrace{\sum_{(i,j) \in L, c} r_{ij}^c \left[ q_i^c(t) - q_j^c(t) \right]}_{\text{should maximize this to minimize the drift.}}$$

- Give $\vec{r} = g(\vec{P}, K(t))$, we have $\sum_c r_{ij}^c = r_{ij}$

$\Rightarrow$ maximizing the last term implies that we should only use $r_{ij}^c$ with the largest value of $(q_i^c(t) - q_j^c(t))$

- The last-term becomes $\sum_{ij} r_{ij} \max_c [q_i^c(t) - q_j^c(t)]$

$\Rightarrow$ should then choose $\vec{P}(t)$ to maximize this weighted sum.

---

## Joint Routing & Scheduling Algo.

① Maximum differential backlog.

For each link $(i,j)$, select the commodity $c$ such that the value

$$q_i^c(t) - q_j^c(t)$$

is the largest.

Let $c_{ij}^*(t) = \arg\max_c q_i^c(t) - q_j^c(t)$

Let $W_{ij}(t) = q_i^{c_{ij}^*(t)} - q_j^{c_{ij}^*(t)}$ denote the maximum differential backlog.

② Scheduling.

Choose $\vec{P}(t)$ such that

$$\vec{P}(t) = \arg\max_{\substack{\vec{P} \in \Pi \\ \vec{r} = g(\vec{P}, K(t))}} \sum_{(i,j)} W_{ij}(t) \cdot r_{ij}$$

Let $\vec{r}(t) = g(\vec{P}(t), K(t))$

③ Routing:

On each link $(i,j)$, route the commodity $c_{ij}^*(t)$ using the rate $r_{ij}$ i.e.

$$r_{ij}^c(t) = \begin{cases} r_{ij}(t) & \text{if } c = c_{ij}^*(t) \\ 0 & \text{, otherwise} \end{cases}$$

Intuition: As packets are queued, the queue difference forms a "gradient", which points to the optimal direction to forward packets

Can be shown to achieve the largest set of offered loads $[\lambda_i^c]$.

Reference: Neely & Modiano.

*Dynamic Power Allocation and Routing for Time Varying Wireless Networks,* by M. J. Neely, E. Modiano and C. E. Rohrs, in IEEE INFOCOM, April 2003.

35

# Back-pressure - handout

Friday, February 01, 2008 3:40 PM

— Each node maintains multiple queues, one for each destination $c$

$q_i^c$

Let $r_{ij}^c$ be the amount of packets of commodity $c$ that are forwarded from node $i$ to node $j$.

$$q_i^c(t+1) = \begin{cases} \left[ q_i^c(t) + \sum_{j \neq i} r_{ji}^c(t) + \lambda_i^c - \sum_{j \neq i} r_{ij}^c(t) \right] & \text{if } i \neq d(c) \\ = 0 & \text{if } i = d(c). \end{cases}$$

---

— We now see why working with the Lyapunov function also produces a joint routing & scheduling policy that is throughput-optimal

— Use the Lyapunov function

$$V(\vec{q}) = \frac{1}{2} \sum_{i,c} (q_i^c)^2$$

Note

$$\left(q_i^c(t+1)\right)^2 - \left(q_i^c(t)\right)^2$$

$$\leq 2 q_i^c(t) \left[ \sum_{j \neq i} r_{ji}^c(t) + \lambda_i^c - \sum_{j \neq i} r_{ij}^c(t) \right] + \text{constant}$$

$$\Rightarrow V(\vec{q}(t+1)) - V(\vec{q}(t))$$

- Given $\vec{r} = g(\vec{P}, K(t))$, we have $\sum_c r_{ij}^c = r_{ij}$

  $\Rightarrow$ maximizing the last term implies that we should only use $r_{ij}^c$ with the largest value of $(q_i^c(t) - q_j^c(t))$

- The last-term becomes

  $\Rightarrow$ should then choose $\vec{P}(t)$ to maximize this weighted sum.

---

## Joint Routing & Scheduling Algo.

① Maximum differential backlog.

For each link $(i,j)$, select the commodity $c$ such that the value

$$q_i^c(t) - q_j^c(t)$$

is the largest.

Let $c_{ij}^*(t) = \arg\max_c q_i^c(t) - q_j^c(t)$

Let $W_{ij}(t) = q_i^{c_{ij}^*(t)} - q_j^{c_{ij}^*(t)}$ denote the maximum differential backlog.

② Scheduling.

Choose $\vec{P}(t)$ such that

$$\vec{P}(t) = \arg\max_{\substack{\vec{P} \in \Theta \\ \vec{r} = g(\vec{P}, K(t))}} \sum_{(i,j)} W_{ij}(t) \cdot r_{ij}$$

Let $\vec{r}(t) = g(\vec{P}(t), K(t))$

(3) Routing :

On each link $(i,j)$, route the commodity $c_{ij}^*(t)$ using the rate $r_{ij}$ i.e.

$$r_{ij}^c(t) = \begin{cases} r_{ij}(t) & \text{if } c = c_{ij}^*(t) \\ 0 & \text{, otherwise} \end{cases}$$

*Dynamic Power Allocation and Routing for Time Varying Wireless Networks,* by M. J. Neely, E. Modiano and C. E. Rohrs, in IEEE INFOCOM, April 2003.

35

# Opportunistic scheduling - 10min

Tuesday, April 08, 2008 12:43 AM

Let $U_i(t)$ = utility of user $i$ at time slot $t$ if it is scheduled

Let $\vec{U} = [U_1, \cdots, U_N]$

- Define a policy $Q$ as a mapping from the utility vector $\vec{U}$ to the index set $\{1, 2, \cdots N\}$

Assume that $Q$ is time-invariant. Given $\vec{U}(t)$, $Q(\vec{U}(t))$ is the index of the user that is scheduled at the slot $t$

---

Objective:

- maximize the average utility in the system
- subject to the fairness constraints that the long-term fraction of time assigned to user $i$ must be no smaller than $r_i$.

where $\sum_i r_i \le 1$

$$\max_{a(\cdot)} \quad E(u_{a(\vec{u})}) = E\left[\sum_{i=1}^{N} u_i \mathbb{1}_{\{a(\vec{u})=i\}}\right]$$

$$\text{sub to} \quad P(a(\vec{u})=i) \geq r_i$$

10

Equivalent formulation - 5min

Sunday, February 16, 2020 4:13 PM

- Suppose $\vec{u}(t)$ can take one of $K$ values
- The $k$-th value is

$$[u_1^k, u_2^k, \ldots u_N^k] \quad \text{—— channel state}$$

- The corresponding probability is $\lambda_k$
- Let $p_i^k$ be the probability of choosing user $i$ at state $k$.

$$\sum_i p_i^k = 1$$

- Then

$$\begin{aligned} \max \quad & \sum_k \lambda_k \sum_i p_i^k u_i^k \\ \text{sub to} \quad & \sum_k \lambda_k p_i^k \geq r_i \\ & \sum_i p_i^k = 1 \quad \text{for all } k. \end{aligned}$$

---

**Lyapunov function** with virtual queue:

- Define a deficit queue

$$q_i(t+1) = \left[ q_i(t) + r_i - 1_{\{Q(\vec{u}(t)) = i\}} \right]^+$$

- $q_i(t) \uparrow$ if the fraction of time that user $i$ is scheduled is less than $r_i$

- $$V(\vec{q}(t)) = \frac{1}{2} \sum_i q_i^2(t)$$

- $$\begin{aligned} & q_i^2(t+1) \\ \leq & \left( q_i(t) + r_i - 1_{\{Q(\vec{u}(t)) = i\}} \right)^2 \\ = & \; q_i^2(t) + 2 q_i(t) \left( r_i - 1_{\{Q(\vec{u}(t)) = i\}} \right) + M_i \end{aligned}$$

$$\Rightarrow \quad V(\vec{q}(t+1)) - V(\vec{q}(t))$$

$$\leq \sum_i q_i(t) r_i - \sum_i q_i(t) 1\{Q(\vec{u}(t)) = i\} + \frac{1}{2}\sum_i M_i$$

---

## Drift + Penalty

- Add penalty

$$\Delta(t) = \sum_K \lambda_K \sum_i p_i^{k,*} U_i^k - \sum_i U_i(t) 1\{Q(\vec{u}(t)) = i\}$$

- Drift + Penalty

$$V(\vec{q}(t+1)) - V(\vec{q}(t)) + \alpha \Delta(t)$$

$$= \sum_i q_i(t) r_i + \sum_K \lambda_K \sum_i p_i^{k,*} U_i^k + \frac{1}{2}\sum_i M_i^2$$

$$- \sum_i 1\{Q(\vec{u}(t)) = i\} \underbrace{\left[ q_i(t) + \alpha U_i(t) \right]}_{\text{choose the user with the largest } q_i(t) + \alpha U_i(t)}$$

---

## Why does it work?

- Can show that

$$E\left[ V(\vec{q}(t+1)) - V(\vec{q}(t)) + \alpha\Delta(t) \right] \leq \frac{1}{2}\sum_i M_i$$

- Summing $t = 0, 1, \cdots, T$ and divided by $T$

$$\frac{E[V(\vec{q}(t+1))] - E[V(\vec{q}(0))]}{T}$$

$$+ \alpha \cdot \left( \sum_K \lambda_K \sum_i U_i^k p_i^{k,*} - \frac{1}{T}\sum_{t=1}^{T}\sum_i U_i(t) 1\{Q(\vec{u}(t)) = i\} \right)$$

$$\leq \frac{1}{2}\sum_i M_i^2$$

$$\Rightarrow \frac{1}{T}\sum_t \sum_i U_i(t) 1\{Q(\vec{u}(t))=i\}$$

$$\geq \sum_k \lambda_k \sum_i U_i^k p_i^{k,*} + \underbrace{\frac{E[V(\vec{q}(t+1))] - E[V(\vec{q}(0))]}{\alpha T}}_{\to 0 \text{ as } T\to+\infty} - \frac{\frac{1}{2}\sum_i M_i^2}{\alpha}$$

- Hence, as $\alpha \to +\infty$, the total utility is close to optimal.

- Further, without $[\cdot]^+$

$$q_i(t+1) \geq q_i(t) + r_i - 1\{Q(\vec{u}(t))=i\}$$

$$\Rightarrow \quad q_i(T) \geq q_i(0) + T r_i - \sum_t 1\{Q(\vec{u}(t))=i\}$$

$$\Rightarrow \quad \frac{1}{T}\sum_t 1\{Q(\vec{u}(t))=i\} \geq r_i + \underbrace{\frac{q_i(0)-q_i(T)}{T}}_{\to 0 \text{ as } T\to+\infty}$$

# Opportunistic scheduling --- handout

Sunday, February 16, 2020 4:13 PM

## Lyapunov function with virtual queue:

- Define a deficit queue

$$q_i(t+1) = \left[q_i(t) + r_i - 1_{\{Q(\vec{u}(t)) = i\}}\right]^+$$

  - $q_i(t) \uparrow$ if the fraction of time that user $i$ is scheduled is less than $r_i$

- $$V(\vec{q}(t)) = \frac{1}{2} \sum_i q_i^2(t)$$

- $$q_i^2(t+1) \le \left(q_i(t) + r_i - 1_{\{Q(\vec{u}(t)) = i\}}\right)^2$$

$$\Rightarrow \quad V(\vec{q}(t+1)) - V(\vec{q}(t))$$

$$\le \sum_i q_i(t) r_i - \sum_i q_i(t) 1_{\{Q(\vec{u}(t)) = i\}} + \frac{1}{2} \sum_i M_i$$

---

## Drift + Penalty

- Add penalty

$$\Delta(t) =$$

- Drift + Penalty

$$V(\vec{q}(t+1)) - V(\vec{q}(t)) + \alpha \Delta(t)$$

$$=$$

## Why does it work?

- Can show that

$$E\left[V(\vec{q}(t+1)) - V(\vec{q}(t)) + \alpha\Delta(t)\right] \le \frac{1}{2}\sum_i M_i$$

- Summing $t=0,1,\cdots,T$ and divided by $T$

$$\frac{E\left[V(\vec{q}(t+1))\right] - E\left[V(\vec{q}(0))\right]}{T}$$

$$+ \alpha\cdot\left(\sum_k \lambda_k \sum_i U_i^k p_i^{k,*} - \frac{1}{T}\sum_{t=1}^{T}\sum_i U_i(t) 1\{Q(\vec{u}(t))=i\}\right)$$

$$\le \frac{1}{2}\sum_i M_i^2$$

$$\Rightarrow \frac{1}{T}\sum_t\sum_i U_i(t) 1\{Q(\vec{u}(t))=i\}$$

$$\ge \sum_k \lambda_k \sum_i U_i^k p_i^{k,*} + \underbrace{\frac{E\left[V(\vec{q}(t+1))\right] - E\left[V(\vec{q}(0))\right]}{\alpha T}}_{\to 0 \text{ as } T\to+\infty} - \frac{\frac{1}{2}\sum_i M_i^2}{\alpha}$$

- Hence, as $\alpha\to+\infty$, the total utility is close to optimal.

- Further, without $[\cdot]^+$

$$q_i(t+1) \ge q_i(t) + r_i - 1\{Q(\vec{u}(t))=i\}$$

$$\Rightarrow q_i(T) \ge q_i(0) + Tr_i - \sum_t 1\{Q(\vec{u}(t))=i\}$$

$$\Rightarrow \quad \frac{1}{T} \sum_t \mathbb{1}\{a(\vec{u}(t)) = i\} \geq r_i + \underbrace{\frac{q_i(0) - q_i(T)}{T}}_{\to 0 \text{ as } T \to +\infty}$$

# Challenge

Monday, March 01, 2010 12:07 PM

Max-weight policy is powerful in that it deals with arbitrarily settings.
However, it suffers from

- Computational and communication overhead
- No explicit expression for the capacity

An alternative is to make simplifying assumption on the system setting, and aim to gain simpler expressions. This is the case with stochastic geometry and scaling laws.

# Stochastic geometry - 10min

Friday, February 9, 2018 10:49 AM

- Focus on downlink. (Uplink can be treated in an analogous manner.)

- Consider a fixed mobile at the origin

- Key assumption #1: Base-stations (BS) are distributed according to a homogeneous Poisson Point Process (PPP) of intensity $\lambda$.

    - Roughly speaking, in any small area $\Delta A$, the probability of having one BS in the area is $\lambda \Delta A$. independently of other BSs

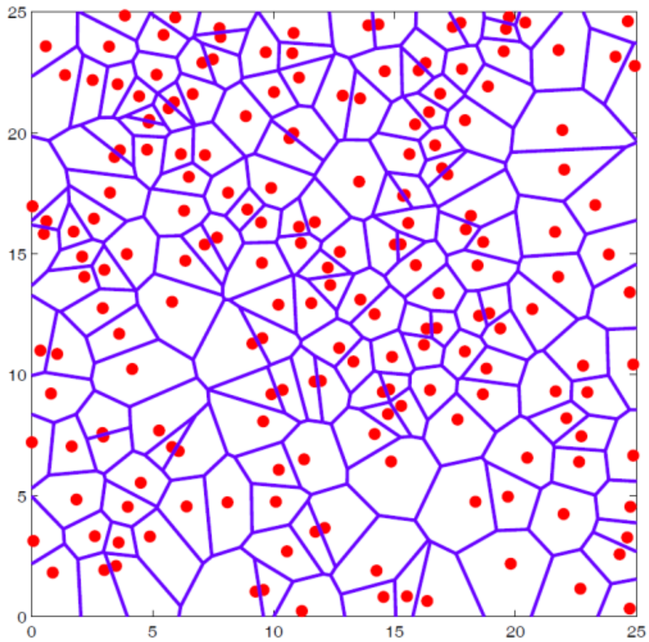
    - When $\Delta A$ is large, the # of BS is a Poisson random variable with mean $\lambda \Delta A$

$$P\{\# \text{ of BS} = k\} = e^{-\lambda \Delta A} \frac{(\lambda \Delta A)^k}{k!} \quad k = 0, 1, \ldots$$

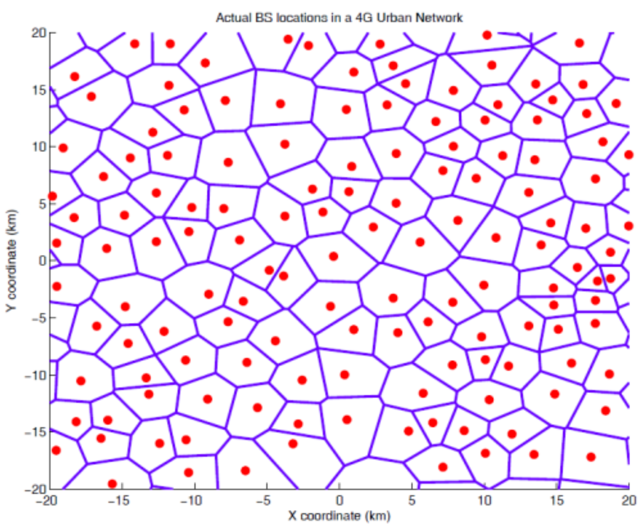
- The independence assumption does not hold when BSs are arranged according to some patterns
    - may be more relevant for small cells in HetNets.

- show Fig. 1 & Fig. 2 in Andrews et al.

In optimal scheduling, we assume that node locations are given.
$\Rightarrow$ difficult to obtain closed form solutions



PPP




Actual BS in 4G networks

Such a PPP assumption allows us to derive key quantities in closed form

---

- The mobile is connected to the nearest BS $b_0$ at a distance $r$.
- How far is $r$? For any value $a$

$$P[r > a] = P\{\# \text{ of } BS = 0 \text{ in area } \pi a^2\}$$

$$= e^{-\lambda \pi a^2}$$

$\Rightarrow$ The pdf of $r$ is

$$\ldots e^{-\lambda \pi r^2}\Big| = 2\pi\lambda r \cdot e^{-\lambda \pi r^2}$$

$\Rightarrow$ The pdf of $r$ is

$$f(r) = \frac{d}{dr}\left[1 - e^{-\lambda\pi r^2}\right] = 2\pi\lambda r \cdot e^{-\lambda\pi r^2}$$

– We then then derive the distribution of received power