

Multi-Hop Routing and Scheduling in

Wireless Networks in the SINR Model

Guy Even

Yakov Matsri

Moti Medina

School of EE, Tel-Aviv Univ

WRAN - Iceland - June 30 2011

# Problem: Throughput Maximization

1)  $n$  nodes in the plane

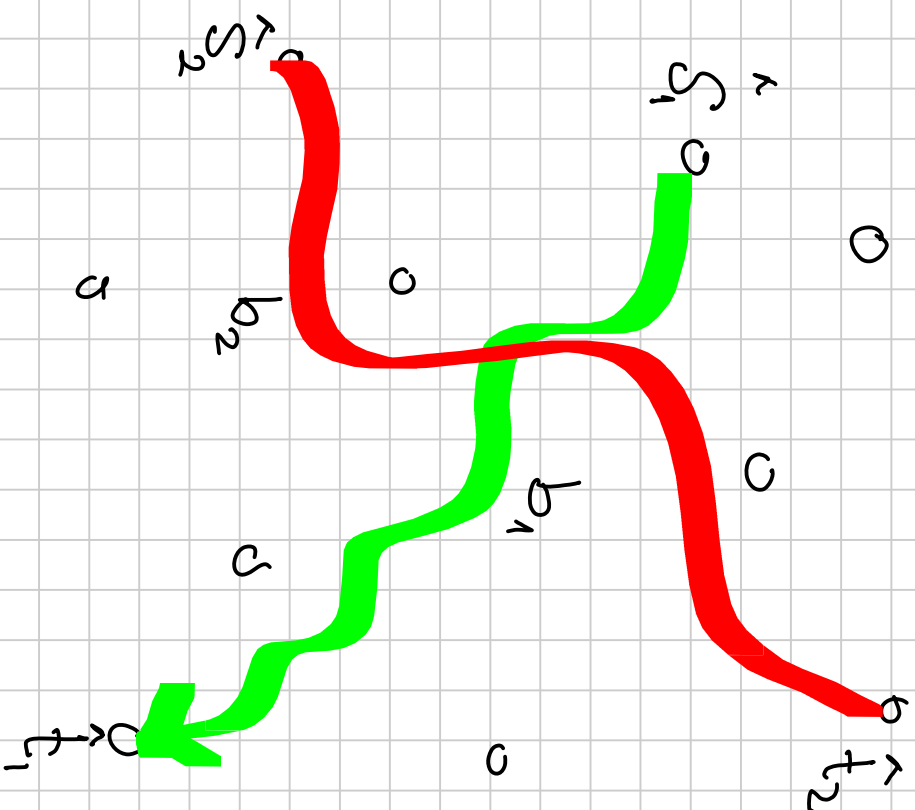
2)  $k$  requests  $R_i = (s_i, t_i, b_i)$

want: schedule packets in network for the requests.

- \* packets may travel along different paths

- \* multi-hop: solve routing

- \* interference model: SINR



## Spec: Output of Algorithm

Output: multi-commodity flow  $f$  & schedule  $\{L_t\}_{t=1}^T$

Multi-commodity flow:  $f = (f_1, \dots, f_k)$ , where  $f_i$  for request  $R_i$

Schedule: Seq. of  $T$  SINR-free sets of links  $\{L_t\}_{t=1}^T$ .

Schedule **supports** flow:

$$\forall e: \frac{f(e)}{c(e)} \leq \frac{1}{T} \mid \{t: e \in L_t\}$$

$$\text{Goals: } \max \sum_{i=1}^T \min \left\{ 1, \frac{|f_i|}{b_i} \right\} \quad (\text{Throughput})$$

$$\max \min_i \frac{|f_i|}{b_i} \quad (\text{max-min throughput})$$

Offline: 1) queue management. 2) compare to shortest schedule.

# Results

who	linear powers	given powers	powers $\in [P_{min}, P_{max}]$
Chafekar et al. Infcom 2008	$\log \Delta$ (also uniform power)	$\log T \cdot \log \Delta$	
Usam et al. Infocom 2011 *	$\log n$ (also unbounded)		$\log T \cdot \log n$
We	$\log n$	$\log n \cdot (\log T + \log \Delta)$	$(\log n + \log \log T) \cdot (\log T + \log \Delta)$

$$\Gamma \stackrel{\Delta}{=} \frac{P_{max}}{P_{min}} \quad \Delta = \frac{d_{max}}{d_{min}}$$

\* Main: "only of theoretical interest and quite infeasible practically"

## Assumptions

- 1) Same SINR threshold for all the links.
- 2) All links use a single mode code.
- 3) uniform fixed packet length.  
 $\Rightarrow$  each link can deliver one packet per slot.
- 4) links bounded away from SINR threshold

$$(*) \quad \frac{S^c}{N} \geq (1+\epsilon) \cdot \beta$$

off line: is (\*) a technicality or a "real" issue

# The Plan

1) Formulate an LP-relaxation

Solution:  $f = (f_1, \dots, f_k)$  that serves  $(R_1, \dots, R_k)$

2) compute an SINR-feasible schedule  $\{t_i\}_{i=1}^T$  that supports (a fraction of)  $f$ .

Offline: distributed vs. centralized

given  $(f, \{t_i\}_{i=1}^T)$ , control is local



## LP-Relaxation

1) multi-commodity flow  $f = (f_1, \dots, f_k)$ .

$f_i$  is a flow for request  $R_i = (s_i, t_i, b_i)$

2) But, must address interferences. How?

graph model:

$$\frac{f(e)}{c(e)} + \sum_{\substack{e' \text{ interferes} \\ \text{with } e}} \frac{f(e')}{c(e')} \leq 1$$

\* pessimistic \* amenable to "greedy" coloring

SINR model?

call acceptance for help!

## Affectance

$$\frac{S_e}{N + I_e} \geq \beta \implies I_{\max}(e) = \frac{S_e}{\beta} - N$$

$$a_e(e) = \frac{S_e e}{I_{\max}(e)}$$

Claim: a subset  $L$  of links is SINR feasible.

$$\Leftrightarrow \forall e \in L: a_L(e) \leq 1$$



LP constraint?

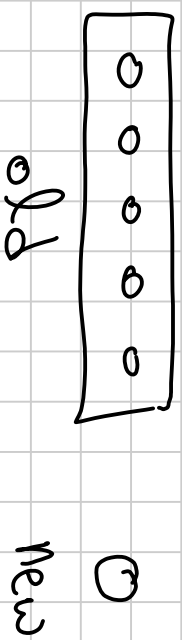
$$f(e) + \sum_{e_i} a_{e_i}(e) \cdot f(e_i) \leq 1$$

1) Not a relaxation! if  $a_{e_i}(e) > 1$  then

$$f(e) = f(e_i) = \frac{1}{2} \text{ is not feas.}$$

$$\text{Solution: } \bar{a}_{e_i}(e) \triangleq \min \{ a_{e_i}(e), 1 \}$$

2) How to schedule?  $\bar{a}_{e_i}(e) \neq \bar{a}_e(e_i)$



$$\begin{aligned} D_{\text{old}}(\text{new}) &\leq 1 \\ \text{but } D_{\text{new}}(e) &> 1 \text{ for } e \in \text{old} \end{aligned}$$

## Buckets

Classify links by **reception** power:

$$B_i = \{ e : 2^i \cdot S_{\min} \leq S_e < 2^{i+1} \cdot S_{\min} \}$$

linear powers: one bucket.

LP relaxation: interference constraint

$$\forall i: \forall e \in B_i: f(e) + \sum_{\{e' \in B_i: d_{e'} \geq d_e\}} (\bar{a}_{e'}(e) + \bar{a}_e(e')) \cdot f(e') \leq 1$$

Remark: Sum over  $d_{e'} \geq d_e$  does not complicate scheduling.

Why is a relaxation? Suppose  $L \in B_i$  is SNR feas.

$$\text{Thm 1: } \forall e \in B_i: \sum_{e' \in L} \bar{a}_{e'}(e) = O(1)$$

$$\text{Thm 2 [K11]: } \forall e \in B_i: \sum_{e' \in L: d_{e'} \geq d_e} \bar{a}_e(e') = O(i)$$

relaxation: use averaging argument ...

## Greedy Coloring

Given flow  $f$  (solution of LP), computes a schedule  $\{L_t\}_{t=1}^T$  that supports  $f/\lambda$  ( $\lambda$  is constant).

Each  $L_t$  satisfies:  $\forall e \in E \sum_{e' \in L_t: d_{e'} \geq d_e} (\bar{a}_{e'}(e) + \bar{a}_e(e')) \leq 1$ .

How? fix  $T$  large enough

1) Scan links in descending length order

2) allocate  $[T \cdot f(e)]$  time slots to  $e$ . (first fit)

always succeeds because  $f(e) + \sum_{e' \in E: d_{e'} \geq d_e} f(e') \leq 1$

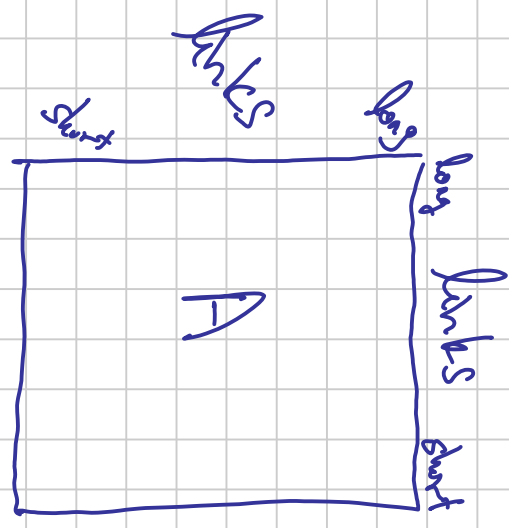
dispersion (making schedule SINR feas)

1) Refine each  $L_t$  into  $O(\log n)$  subsets  $\{L_{t,i}\}$ .

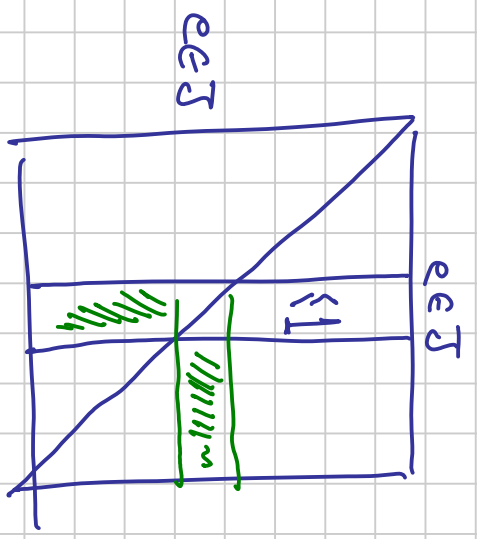
$$\forall e \in L_{t,i} : \bar{a}_{L_{t,i}}(e) \leq 3$$

2) Further refine to obtain SINR-feas  $[H_{t,i}]$ .

$$\sum_{e: d(e) \geq d(e)} (\bar{a}_e(e) + \bar{a}_e(e')) \leq 1 \Rightarrow a_{L_t, i}(e) \leq 3$$

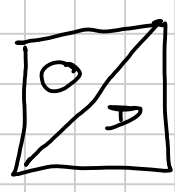


$$A(e', e) = \bar{a}_e(e) + \bar{a}_e(e')$$



we know: in every column

has weight  $\leq 1$ .



$\Rightarrow$  at least half rows have weight  $\leq 2$

denote those by  $J$ .

$\Rightarrow$  weight of col  $e \in J$  in  $A$  is  $\leq 1 + 2 = 3$ .

Each  $T$  contains half the remaining links  
So after  $(\log n)$  iterations, we stop.

More than one bucket?

- deal with each bucket separately.
- concatenate schedules.
- decreases throughput by # buckets.

$$\begin{aligned} \# \text{ buckets} &\leq \log \left( \frac{S_{\max}}{S_{\min}} \right) = \log \left( \frac{P_{\max}}{P_{\min}} \cdot \left( \frac{d_{\max}}{d_{\min}} \right)^{\alpha} \right) \\ &= \log T + \alpha \cdot \log \Delta \end{aligned}$$

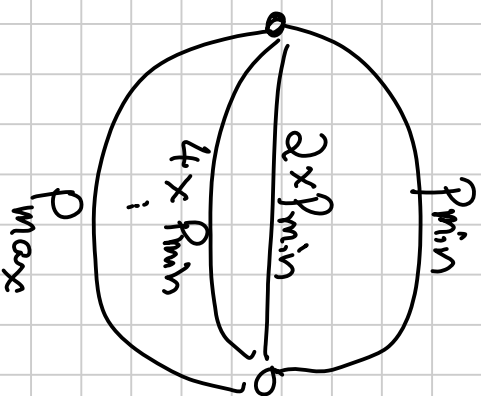
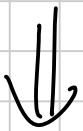
$$\Rightarrow O(\log n \cdot (\log T + \log \Delta)) \text{ approx.}$$



## Limited Powers

each link  $e$  may transmit with power  $P_e \in [P_{\min}, P_{\max}]$ .  
alg needs to compute powers.  
link may change power between time slots.

Reduction



For simplicity, assume:  $P_{\max} = 2^k \cdot P_{\min}$ .

Observation: Limiting opt to discrete powers reduces throughput by a constant.

$$\text{OPT} \rightarrow \text{Heel}_t : P_e^i \leftarrow \min \left\{ 2^i P_{\min} \mid 2^i P_{\min} > P_e \right\}$$

or  $P_e = P_{\max}$

Now  $L_t^{\text{disc}}$  is not SNR feas, but it is a