Wireless networks: Algorithmic trade-offs between Throughput & Delay

Devavrat Shah

Laboratory of Information and Decision Systems
Massachusetts Institute of Technology

Status quo

- Primary performance metric in a wireless network
  - Throughput and delay
  - Necessary for quality-of-service guarantee, buffer-design, etc.
  - Further, algorithm should be implementable (distributed)
- However, thus far most of the work has concentrated on designing throughput optimal algorithms
  - Low delay algorithm design is a lot harder
  - An analogy: being ahead of all in a marathon throughout the race (low delay) versus completing the race first (high throughput)
- One of the main reason for such status
  - Lack of good tools for delay analysis
  - Hence lack of insight about what causes high delay
  - As well as inability to understand finer throughput delay tradeoff

Summary of results

- First, we establish that
  - It is possible to have very simple, distributed throughput optimal algorithm for any network

  $\rightarrow$ throughput is easy

- To understand interaction with throughput and delay
  - We introduce new tools from computational complexity
  - We establish computational impossibility of designing high throughput, low delay algorithm for arbitrary network

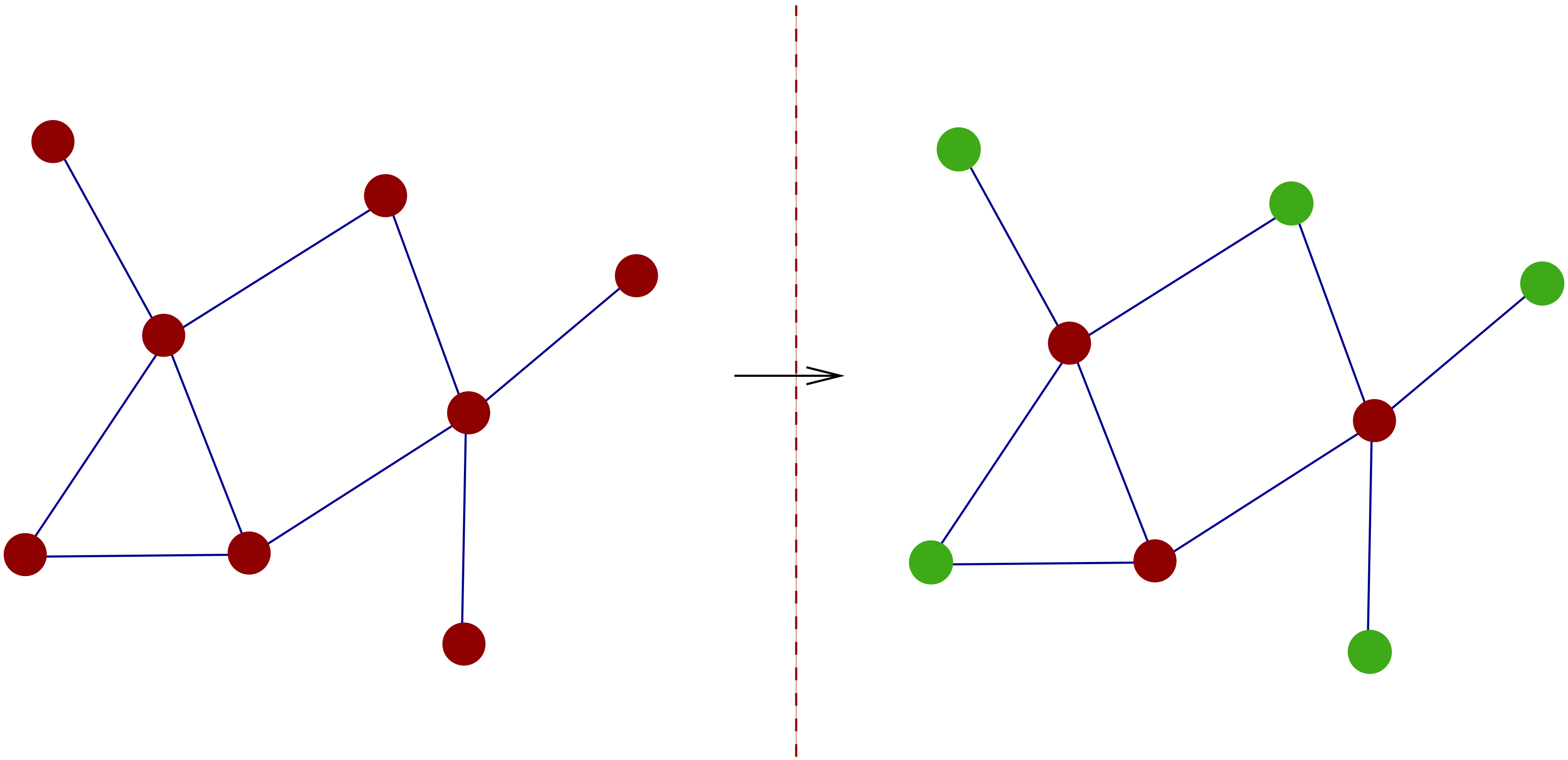
- However, the relevant question is: are practical networks hard ?
  - We obtain novel algorithms using graph theoretic properties of practical networks
    - these are simple, distributed; throughput and delay optimal

Model

- Consider a single-hop wireless network
  - Time is slotted denoted by $t \in \mathbb{Z}$
  - Packets of unit-size arrive at nodes of G
    - At rate λ_v for $v \in V$ according an external arrival process
  - Packets are buffered at nodes if required
    - Let $Q_v(t)$ denote queue-size at $v \in V$ at time t
  - Packets depart from network when transmitted
    - That is, it is a single-hop network

- Scheduling algorithm: at each time
  - Choose an independent set of G
    - To schedule transmission of packets at nodes

Model: An example



Model

- Let $\mathcal{I}(G) \subset \{0,1\}^n$ be set of independent sets of G
  - Let $\Lambda = \text{Co}(\mathcal{I}(G))$

- Capacity region is Λ
  - If $\lambda \notin \Lambda$:
    - Not possible to serve all queues at rate higher than their arrival rate
  - If $\lambda \in \Lambda^o$:
    - Possible to serve all queues at rate higher than arrival rate through a TDMA scheme

$\rightarrow$ λ is admissible if $\lambda \in \Lambda^o$

Performance metric

- Throughput
  - Scheduling algorithm is stable (delivers 100% throughput), if
    - For any admissible λ, the average queue-size is finite

$$\sup_t \mathbb{E}\left[Q_v(t)\right] < \infty, \quad \text{for all} \quad v.$$

- Net average queue-size:
  - $\sup_t \mathbb{E}\left[Q(t)\right]$,
    - where $Q(t) = \sum_v Q_v(t)$

Maximum Weight Scheduling

- Algorithm: max. wt. independent set (MWIS)
  - Every time, choose schedule (independent set) with max. weight,
    - weight of a node is equal to the queue-size
  - Transfer packets according to this schedule

- Tassiulas and Ephremides (1992) proposed this algorithm
  - They showed it to be stable

- It follows that for any G, under max. wt. scheduling
  - The net average queue-size is bounded above as $O(n^2/\varepsilon)$
    - Under Bernoulli i.i.d. arrival process, and
    - $\lambda \in (1-\varepsilon)\Lambda$

Complexity of Max. Wt. Scheduling

- The problem of finding max. wt. independent set is hard
  - There are instances of weighted graphs such that no polynomial in n time algorithm to find even good approximation unless $P = NP$
- Question: is there any algorithm that is stable
  - Requires poly in n computation for any graph G
  - And, is totally distributed
- Answer: Yes
  - Yes, using a *Gossip* mechanism
  - Jung and Shah (2007)

$\longrightarrow$ So what is the issue ?

Complexity of Max. Wt. Scheduling

- These low (poly in n) complexity distributed algorithms
  - Stable, but
  - Induce large (exponential in n) average queue-size

- Question: is there an algorithm that always provides
  - Small (poly in n) avg. queue-size for
    - Bernoulli i.i.d. arrival process with
    - $\lambda \in (1-\varepsilon)\Lambda$ for some fixed $\varepsilon > 0$
  - And has *low (poly in n)* complexity ?

- Answer: No
  - Under standard computational hypothesis
  - Shah and Tsitsiklis (2007)

Good Graph Structure

- Given (family of) graph $G = (V, E)$, $\mathcal{B} \subset V$ be a random set s.t.
  - $\Pr(v \in \mathcal{B}) \leq \varepsilon$ for any $v \in V$
  - $G' = (V \backslash \mathcal{B}, E')$ is made of connected components of diameter Δ

  $\rightarrow$ Called (ε, Δ)-decomposition of G

- A graph is good if it admits (ε, Δ)-decomposition
  - For any $\varepsilon > 0$ and
  - Δ, possibly dependent on ε, but independent of $n = |V|$

- We show that two important class of graphs are good
  - Graphs with low doubling dimension, and
  - Minor excluded graphs

Good Graph Structure

- A graph G has doubling dimension ρ if (essentially)
  - For any $v \in V$, its neighborhood has polynomial growth
  - That is, $\mathbf{B}(v, r) \leq (2r)^{\rho}$ for $r \geq 1$
  - Example: Geometric graphs

- Graph H is a minor of G if
  - H can be obtained from G through an arbitrary sequence operations:
    - edge removal or merging of two connected vertices
  - Example: all Planar graphs

$\longrightarrow$ We have simple, distributed (ε, Δ)-decomposition for these graphs

MWIS for good graph structure

- Given G with non-negative node weights,
  - Obtain (ε, Δ)-decomposition $\mathcal{B}$
  - Set all nodes in $\mathcal{B}$ to 0
  - Let $S_1, \ldots, S_\ell$ be connected components of $G' = (V \backslash \mathcal{B}, E')$
    - compute max. wt. independent set restricted to each S_i
  - Output thus computed assignment of nodes as estimate of max. wt. independent set
    - use it as schedule for wireless transmissions

- Features of algorithm
  - It is totally distributed and linear (in network size n) running time
  - That is, each node performs constant number of operations

Performance of scheduling algorithm

- The algorithm proposed is
  - Throughput optimal, i.e. stable and delay (order) optimal

- **Theorem.** (Jung-Shah '07) The algorithm is stable, induces average queue-size $O(n)$ and takes $O(n)$ overall computation time in a totally distributed manner.

- Summary:
  - Easy to design only throughput optimal distributed algorithm, but
    - impossible to design both throughput optimal and low delay algorithm for arbitrary network
  - However, for most practical networks
    - it is possible to design both throughput and delay optimal distributed algorithms

Goals

- We hope to achieve the following two goals by the end-of-phase
  - **Goal 1.** establish that it is not possible to design computationally efficient high throughput and low delay algorithm for wireless network under physical (SINR) model
  - **Goal 2.** design simple and distributed throughput-delay optimal algorithm for practical wireless network topologies under physical model

Community challenge

- Information theoretic version of throughput versus delay
    - Given a large wireless network, establish that capacity region characterization is computationally hard problem
    - Our methods will imply that it is computationally hard to have high throughput and low delay algorithms
    - Classically, delay is measured in terms of block-length
    - Thus, it will not be possible to design computationally efficient short block-codes