Minimum-Length Scheduling: A Cross-Layer Approach

Anna Pantelidou
Centre for Wireless Communications, University of Oulu

GIGA Seminar 2010

11-01-2010
Introduction

- Scheduling is a fundamental problem in wireless systems
  - **Who** transmits and **when**
  - Depends on transmission powers, **rates**
- Various **performance objectives**
  - Utility Maximization (e.g., Fairness)
  - QoS (e.g., Delay)
  - Throughput (e.g., Stability, Sum Throughput)
- To optimize under these objectives one must assume the channel is stationary and ergodic
- Wireless channel is unpredictable
  - Mobility, nodes with finite energy resources, etc.
- **Minimum-length Scheduling**: alternative performance metric
  - Minimize **time** to meet given demand (rate, volume) at destinations
  - Focus on **volume**
    - Related to throughput maximization
    - Valid also for non-ergodic, non-stationary links
Network Model

- Single-hop multiple-access setting
  - $K$ source/destination pairs
- Slotted time $t \in \{0, 1, \ldots\}$
- Transmission power of source $k$ at slot $t$
  \[ P_k(t) \in \{0, P_k^{\text{max}}\} \]
- Time-invariant channel
  \[ 0 < G(k, j) \leq 1 \]
- Noise power at destination $j : N_j$
- Initial volume at source $k : d_k$
- Queue size of source $k$ at slot $t : X_k(t)$
  - Queue-size is initialized to $X_k(0) = d_k$
Determining Transmission Success

- Source $k$ transmits "successfully" to destination $k$ at rate $r_k$ if

$$\text{SINR} = \frac{P_k(t)G(k,k)}{N_k + \sum_{j=1, j \neq k}^{K} P_j(t)G(j,k)} \geq \theta_{t,k}(r_k)$$

- SINR threshold is a function of various parameters, e.g.,
  - Target error rate
  - Modulation
  - Coding
  - Transmission rate (bits/sec)

- SINR threshold is an increasing function of the maximum rate
A Fundamental Trade-Off

- Decreasing transmission rate ➔ Decreases threshold value
  - More sources jointly satisfy the SINR criterion
  - But sources transmit for a higher fraction of the time
- Increasing transmission rate ➔ Increases threshold value
  - Fewer sources jointly satisfy the SINR criterion
  - More time-sharing needed which decreases the effective rate
- Which mode of operation is preferable?
  - Depends on performance objective and channel conditions
Rate Control and Scheduling Policy

- $\mathcal{R}$: Set of all feasible $K$-dimensional rate vectors
  - Set with cardinality: $|\mathcal{R}| = 2^K - 1$

- Centralized network control policy selects at every slot a rate vector
  $$\mathbf{r}(t) = (r_1(t), \ldots, r_K(t)) \in \mathcal{R}$$
  - All sources $k$ for which $r_k(t) = 0$ are not scheduled

- Queue size evolution $\mathbf{X}(t+1) = [\mathbf{X}(t) - \mathbf{r}(t+1)]^+$
  - Queue sizes keep decreasing until they empty

- **Objective:**
  - Select a **rate vector** at each slot to **minimize** the **number of slots** until all queues empty
An Example

- Vector of initial demands \( \mathbf{X}(0) = \begin{bmatrix} 4 \\ 6 \end{bmatrix} \)
- Rate vectors

\[
\begin{align*}
\mathbf{r}^1 &= \begin{bmatrix} 3 \\ 0 \end{bmatrix}, & \mathbf{r}^2 &= \begin{bmatrix} 0 \\ 3 \end{bmatrix}, & \mathbf{r}^3 &= \begin{bmatrix} 2 \\ 2 \end{bmatrix}
\end{align*}
\]
An Example

- Each vertex represents a queue size vector
- Each directed edge corresponds to a rate vector in $\mathcal{R}$
- Weight of an edge equals one (time slot)

$$ r^1 = \begin{bmatrix} 3 \\ 0 \end{bmatrix} $$
An Example

\[
\mathbf{r}^2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}
\]
An Example

- For each vertex
  - $|\mathcal{R}|$ potentially outgoing edges
  - Each edge $\mathbf{r}^i, i = 1, \ldots, |\mathcal{R}|$
    - Incident from $\mathbf{x}_i$
    - Incident to $\mathbf{y}_i = [\mathbf{x}_i - \mathbf{r}^i]^+$

\[
\mathbf{r}^3 = \begin{bmatrix}
2 \\
2
\end{bmatrix}
\]
An Example

\[ \mathbf{r}^1 = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \]
An Example

\[
\begin{align*}
\mathbf{r}^1 &= \begin{bmatrix} 1 \\ 6 \end{bmatrix} \\
\mathbf{r}^2 &= \begin{bmatrix} 0 \\ 6 \end{bmatrix} \\
\mathbf{r}^3 &= \begin{bmatrix} 2 \\ 4 \end{bmatrix}
\end{align*}
\]

\[
\mathbf{r}^2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}
\]
An Example

\[
\begin{align*}
\mathbf{r}^3 &= \begin{bmatrix}
2 \\
2
\end{bmatrix}
\end{align*}
\]
An Example
An Example
An Example

[Diagram with matrices and vectors]

\[
\begin{bmatrix}
4 \\
0 \\
\end{bmatrix}, \quad
\begin{bmatrix}
0 \\
6 \\
\end{bmatrix}, \quad
\begin{bmatrix}
1 \\
6 \\
\end{bmatrix}, \quad
\begin{bmatrix}
4 \\
3 \\
\end{bmatrix}, \quad
\begin{bmatrix}
1 \\
3 \\
\end{bmatrix}, \quad
\begin{bmatrix}
0 \\
3 \\
\end{bmatrix}, \quad
\begin{bmatrix}
1 \\
0 \\
\end{bmatrix}, \quad
\begin{bmatrix}
0 \\
1 \\
\end{bmatrix}, \quad
\begin{bmatrix}
0 \\
0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
4 \\
6 \\
\end{bmatrix}, \quad
\begin{bmatrix}
2 \\
4 \\
\end{bmatrix}, \quad
\begin{bmatrix}
0 \\
4 \\
\end{bmatrix}, \quad
\begin{bmatrix}
2 \\
1 \\
\end{bmatrix}, \quad
\begin{bmatrix}
2 \\
0 \\
\end{bmatrix}, \quad
\begin{bmatrix}
0 \\
2 \\
\end{bmatrix}
\]
An Example
An Example
Combinatorially Complex Problem

- Minimum-length scheduling is a shortest path problem
- For a Directed Acyclic Graph \( G = (V, E) \)
  - Overall running time of finding a shortest path is \( \Theta(|V| + |E|) \)
  - \( |V| \) and \( |E| \) grow exponentially
  - As the number of transmitters increases
  - As initial demands grow

- Minimum-length scheduling
  - Hard **discrete optimization problem**
  - Solution alternatives
    - Brute force
    - Heuristics
    - Search space **reduction**
Time-Varying Channels (Perfect CSI)

**Assumptions**
- Channel evolves according to a finite state Markov Chain
- Network control policy
  - Knows the channel state at each slot
  - Takes decisions based on current channel state and queue sizes

**Objective**
- Minimize expected number of slots to empty all queues at sources
- Problem can be formulated through Stochastic Shortest Paths (special case of Markov Decision Processes - MDPs)
  - **State Space:** Set of pairs of channel states and queue sizes
    - **Terminating States:** States corresponding to all queues being empty
  - **Action Space:** Set of $2^K - 1$ rate control and scheduling vectors
  - **Cost:** Unitary cost for taking any action from non-terminating states
  - **System Dynamics:** Depend on transition probability of channel process
- Optimal solution satisfies Bellman’s equations
Numerical Results

- 2 source/destinations
- Feasible rates
  - Good state:
    \[
    \begin{bmatrix}
    3 \\ 0 \\
    \end{bmatrix}
    \begin{bmatrix}
    0 \\ 3 \\
    \end{bmatrix}
    \begin{bmatrix}
    2 \\
    2 \\
    \end{bmatrix}
    \]
  - Bad state:
    \[
    \begin{bmatrix}
    2 \\ 0 \\
    \end{bmatrix}
    \begin{bmatrix}
    0 \\ 2 \\
    \end{bmatrix}
    \begin{bmatrix}
    1 \\
    1 \\
    \end{bmatrix}
    \]
Time-Varying Channels (Imperfect CSI)

- **Assumptions**
  - Channel evolves according to a finite state Markov Chain
  - Network control policy
    - Knows the probability distribution of the channel process
    - Transmissions fail due to the uncertainty about the exact channel state
    - Policy has access to feedback regarding the transmission outcome

- **Objective**
  - Minimize expected number of slots to empty all queues at sources
  - Problem can be solved through Partially Observable MDPs
    - **State Space**: Set of pairs of channel states and queue sizes
      - **Terminating States**: States corresponding to all queues being empty
    - **Action Space**: Set of rate control and scheduling vectors corresponding to all possible channel realizations
    - **Observations** $\{0, 1\}$: Indicate the outcome of a transmission
    - **Cost**: Unitary cost for taking any action from non-terminating states
    - **System Dynamics**: Depend on transition probability of channel process and observation probabilities
Numerical Results

- **Solution**
  - Map the problem to one with full observability
  - Solution through dynamic programming
  - Computing the optimal policy is prohibitive
    - Heuristics are necessary

- **Heuristics**
  I. Maximum Likelihood Heuristic
     \[ \pi_{MLH}(\omega(t)) = \pi^*_{MDP}(\arg\max_{\ell \in S}(\omega_\ell(t))) \]
  II. Voting Heuristic
     \[ \pi_{VH}(\omega(t)) = \arg\max_r \sum_{\ell \in S} \omega_\ell(t) \delta(\pi^*_\ell - r) \]

where \( \omega_j(t) \) is the conditional probability that the system is in state \( j \) given the set of actions and observations up to slot \( t \)
Concluding Remarks

- Considered the problem of minimum-length scheduling
  - For stationary channels
  - Time-varying channels under perfect channel state information
  - Time-varying channels under imperfect channel state information
- Minimum-Length Scheduling is a useful alternative objective also for non-ergodic, non-stationary environments
- Tightly related to throughput maximization
- Possible generalizations
  - Multi-hop networks
  - Decentralized algorithms
  - Tackle non-stationarity and non-ergodicity