E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

Solution

1. Messages longer than x can be ignored as they can never be intercepted

E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

Solution

1. Messages longer than x can be ignored as they can never be intercepted
   - Mostly - final result can not be smaller than the longest message!

E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

Solution

1. Messages longer than x can be ignored as they can never be intercepted
   - Mostly - final result can not be smaller than the longest message!
2. Solve the case if at most 1 message was allowed to be intercepted

E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

Solution

1. Messages longer than x can be ignored as they can never be intercepted
   - Mostly - final result can not be smaller than the longest message!
2. Solve the case if at most 1 message was allowed to be intercepted
3. Generalize it to at most 2 intercepted messages

E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

Solution if at most 1 message is allowed to be intercepted

1. If a message (regardless of length) starts at time a, the *next* message can start no earlier than time $a + x + 1 - t$, where t is the length of the *next* message.

E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

Solution if at most 1 message is allowed to be intercepted

1. If a message (regardless of length) starts at time a, the *next* message can start no earlier than time $a + x + 1 - t$, where t is the length of the *next* message.
2. Repeating this, the total time to send all n messages becomes

$$x + 1 + \sum_{i=2}^{n-1} (x + 1 - t_i),$$

where t_i is the length of the ith message sent.

E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

Solution if at most 1 message is allowed to be intercepted

1. If a message (regardless of length) starts at time a, the *next* message can start no earlier than time $a + x + 1 - t$, where t is the length of the *next* message.
2. Repeating this, the total time to send all n messages becomes

$$x + 1 + \sum_{i=2}^{n-1} (x + 1 - t_i),$$

where t_i is the length of the ith message sent.

3. Note: length of the first and last message does not affect the total send time!

E — Eavesdropper Evasion

Problem

Find a way to schedule sending n messages as fast as possible but so that at most 2 of them are fully sent within any time frame of length x.

Solution if at most 1 message is allowed to be intercepted

1. If a message (regardless of length) starts at time a, the *next* message can start no earlier than time $a + x + 1 - t$, where t is the length of the *next* message.
2. Repeating this, the total time to send all n messages becomes
$$x + 1 + \sum_{i=2}^{n-1}(x + 1 - t_i),$$
where t_i is the length of the ith message sent.
3. Note: length of the first and last message does not affect the total send time!
4. By placing the two shortest messages first and last, we get the optimal solution.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1}(x + 1 - t_i)$.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1}(x + 1 - t_i)$.
   - Take the 4 shortest messages and put first and last in each channel.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1}(x + 1 - t_i)$.
   - Take the 4 shortest messages and put first and last in each channel.
   - Each remaining message of length t gets a weight $w = x + 1 - t$.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1} (x + 1 - t_i)$.
   - Take the 4 shortest messages and put first and last in each channel.
   - Each remaining message of length t gets a weight $w = x + 1 - t$.
3. We have $m = n - 4$ weights $w_1, \ldots, w_m$ of total weight $W = w_1 + \ldots + w_n$.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1}(x + 1 - t_i)$.
   - Take the 4 shortest messages and put first and last in each channel.
   - Each remaining message of length t gets a weight $w = x + 1 - t$.
3. We have $m = n - 4$ weights $w_1, \ldots, w_m$ of total weight $W = w_1 + \ldots + w_n$.
4. Want to find a subset $S \subseteq [m]$ such that $\sum_{i \in S} w_i$ is as close to $W/2$ as possible.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1}(x + 1 - t_i)$.
   - Take the 4 shortest messages and put first and last in each channel.
   - Each remaining message of length t gets a weight $w = x + 1 - t$.
3. We have $m = n - 4$ weights $w_1, \ldots, w_m$ of total weight $W = w_1 + \ldots + w_n$.
4. Want to find a subset $S \subseteq [m]$ such that $\sum_{i \in S} w_i$ is as close to $W/2$ as possible.
5. This is a Subset Sum/Knapsack problem.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1}(x + 1 - t_i)$.
   - Take the 4 shortest messages and put first and last in each channel.
   - Each remaining message of length t gets a weight $w = x + 1 - t$.
3. We have $m = n - 4$ weights $w_1, \ldots, w_m$ of total weight $W = w_1 + \ldots + w_n$.
4. Want to find a subset $S \subseteq [m]$ such that $\sum_{i \in S} w_i$ is as close to $W/2$ as possible.
5. This is a Subset Sum/Knapsack problem.
   - Classic dynprog algorithm solves it in time $O(nW) = O(n^2 x)$. Too slow.

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1}(x + 1 - t_i)$.
   - Take the 4 shortest messages and put first and last in each channel.
   - Each remaining message of length t gets a weight $w = x + 1 - t$.
3. We have $m = n - 4$ weights $w_1, \ldots, w_m$ of total weight $W = w_1 + \ldots + w_n$.
4. Want to find a subset $S \subseteq [m]$ such that $\sum_{i \in S} w_i$ is as close to $W/2$ as possible.
5. This is a Subset Sum/Knapsack problem.
   - Classic dynprog algorithm solves it in time $O(nW) = O(n^2 x)$. Too slow.
   - Can be solved in $O(n \cdot x)$ time (Pisinger, 1999).

E — Eavesdropper Evasion

Solution for the 2-message case

1. Key observation: we can view this as having 2 separate channels, and we want at most 1 intercepted message in each channel (not immediately obvious!).
2. Only thing to decide is which messages to send on which channel.
   - Channel with r msgs of lengths $t_1, \ldots, t_r$ finishes in time $x + 1 + \sum_{i=2}^{r-1}(x + 1 - t_i)$.
   - Take the 4 shortest messages and put first and last in each channel.
   - Each remaining message of length t gets a weight $w = x + 1 - t$.
3. We have $m = n - 4$ weights $w_1, \ldots, w_m$ of total weight $W = w_1 + \ldots + w_n$.
4. Want to find a subset $S \subseteq [m]$ such that $\sum_{i \in S} w_i$ is as close to $W/2$ as possible.
5. This is a Subset Sum/Knapsack problem.
   - Classic dynprog algorithm solves it in time $O(nW) = O(n^2x)$. Too slow.
   - Can be solved in $O(n \cdot x)$ time (Pisinger, 1999).

Statistics at 4-hour mark: 19 submissions, 0 accepted