13th Annual CubeSat Developer's Workshop

April 20-22, 2016

San Luis Obispo, CA

Online Resource Allocation and Scheduling for Store-and-Forward Communications with Priority Levels in Nanosatellite Systems

Cherry Y. Wakayama

Peter J. Yoo

Zelda B. Zabinsky

SPAWAR Systems Center Pacific

San Diego, CA

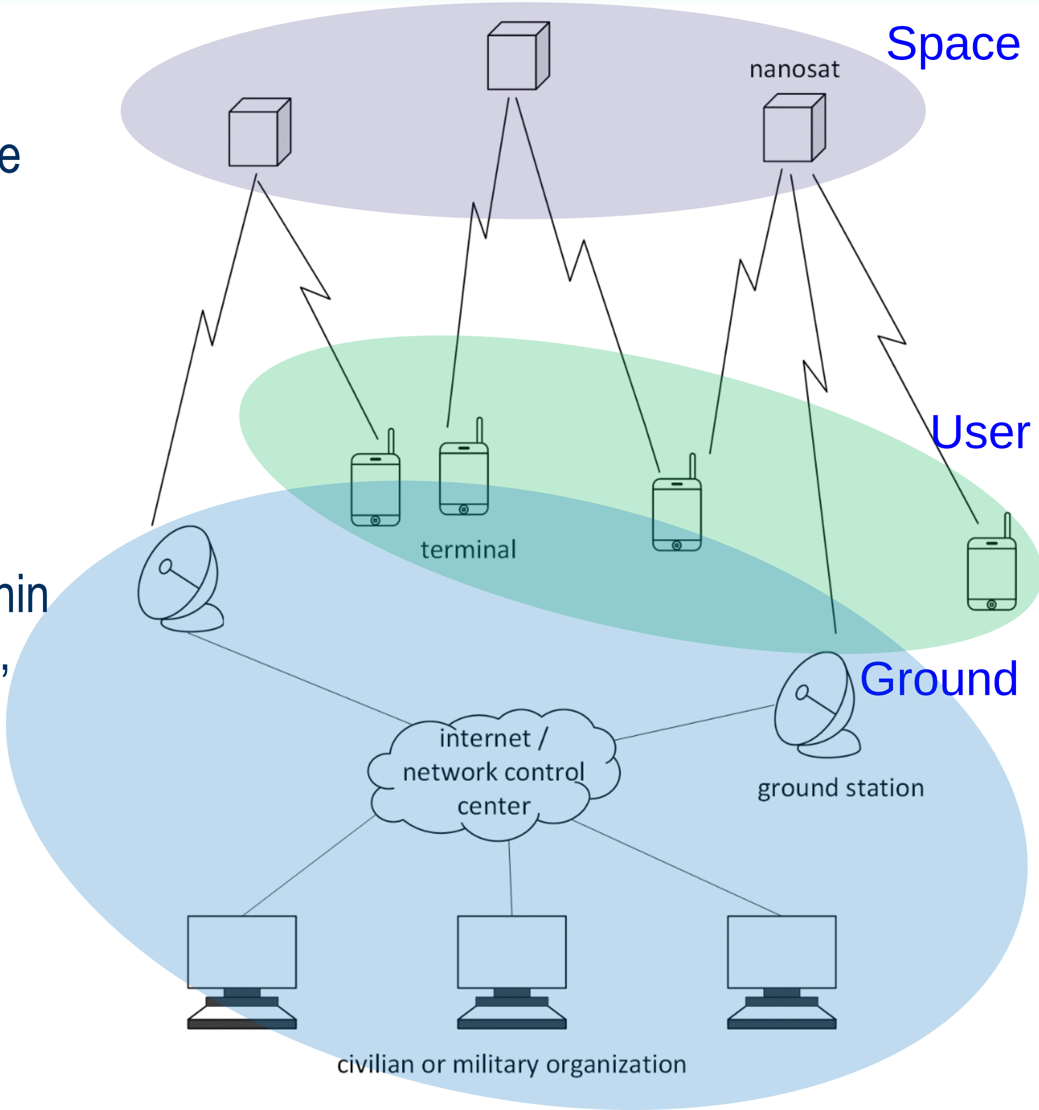
Motivation

Goal:

Provide network connectivity in the remote and hard-to-reach areas using a nanosat constellation

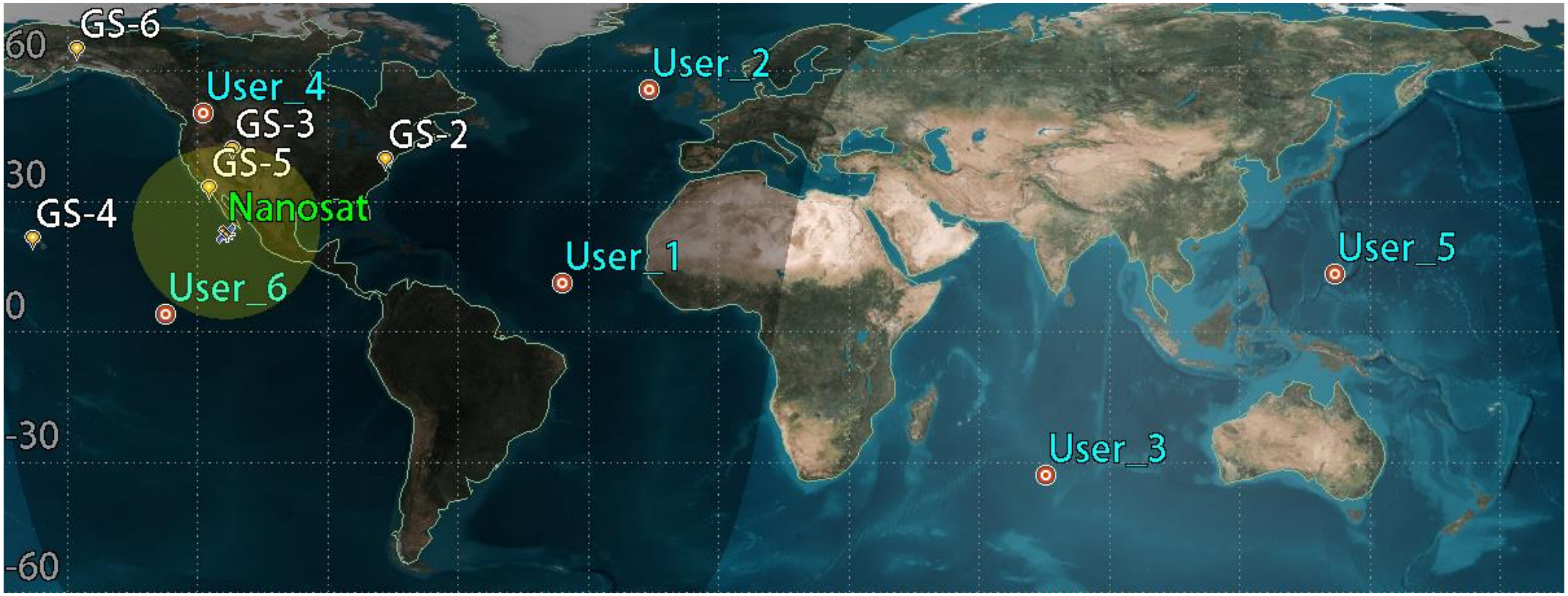
Problem:

How to effectively schedule upload and download events (which, what, when) within the resource constraints (time, bandwidth, energy)?

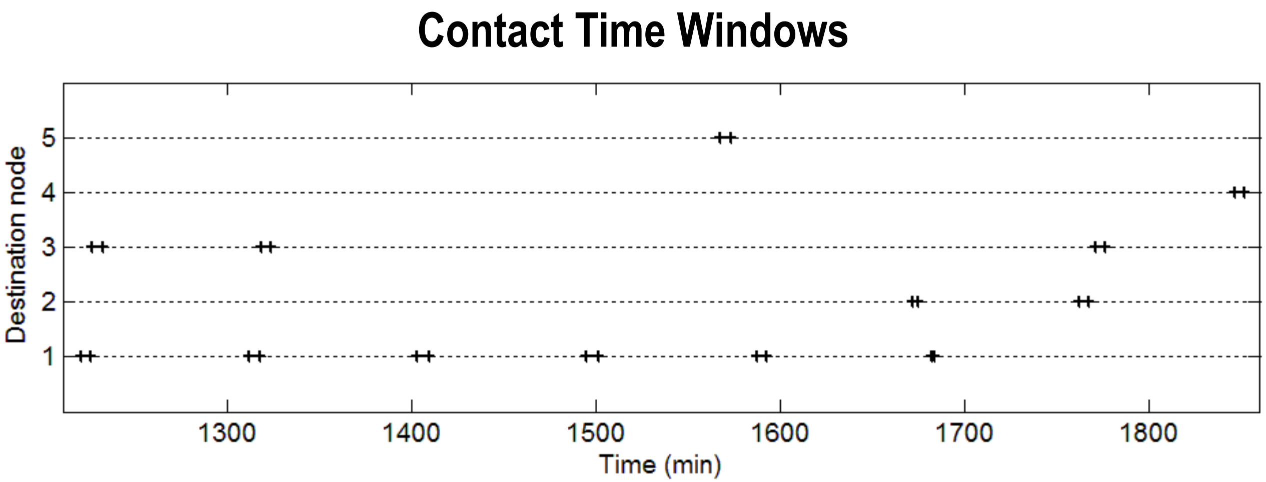


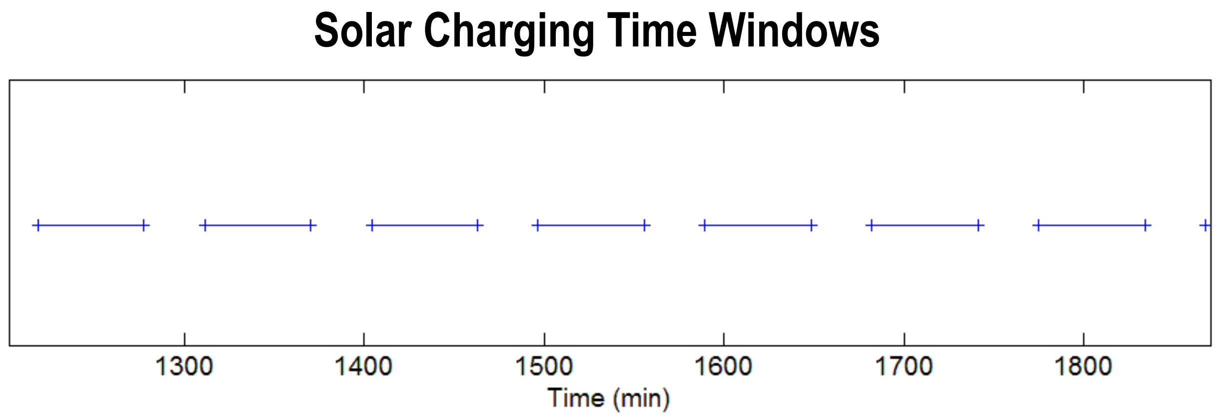
Nanosat Scheduling Problem Motivation

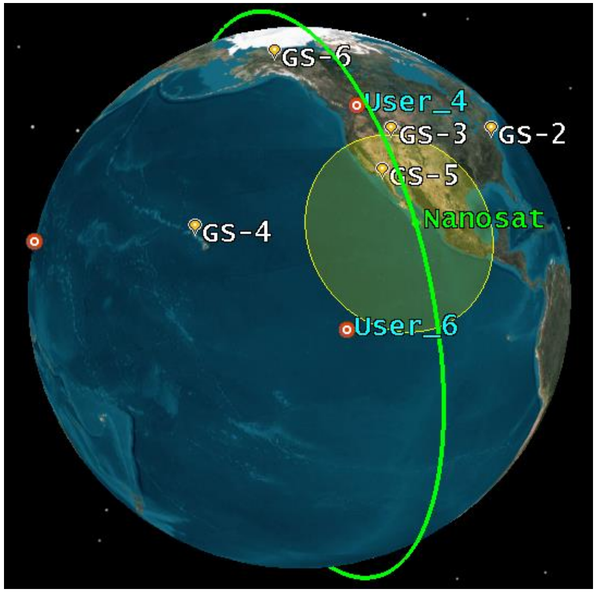
- Messages are collected at ground stations and terminals in remote areas.
- Messages are sent to other ground stations or terminals using nanosats.
- Some messages are time-sensitive or high-priority.
- Network users want the messages as soon as possible.
- No crosslinks among nanosats are assumed.



Nanosat Scheduling Problem with Time Windows







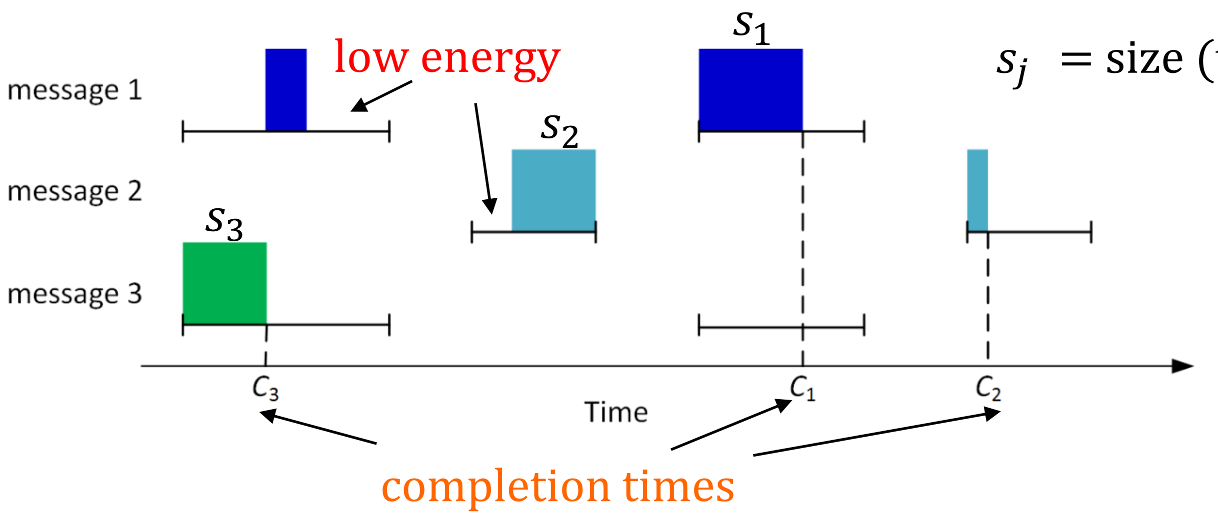
Why is this a new Scheduling Problem?

- Minimizing weighted completion time rather than maximizing data download

$$\min \sum_{j \in J} w_j C_j$$

J = set of messages
w_j = priority weight
C_j = completion time

- Contact windows, charging windows & limited energy storage
- Preemptive scheduling – messages may be forwarded in multiple passes



s_j = size (transmission duration)

Nanosat Scheduling Problem Characteristics

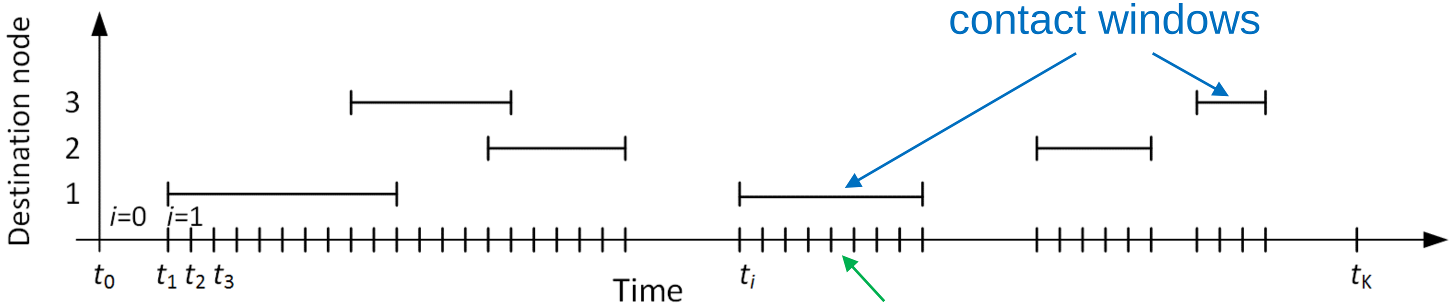
- Single machine scheduling with release dates ($1|r_j|\Sigma w_j C_j$) and additional energy constraints
- ($1|r_j|\Sigma w_j C_j$) is proven NP-hard (No known polynomial-time algorithm).
- Many single machine scheduling problems are solved by creating rules which are optimal for simple problems with structures, but nonetheless useful to solving complex real-world problems
  - Shortest processing time: Optimal for minimizing the average flowtime
  - Earliest due date: Optimal for minimizing the maximum tardiness

Our solution approach

- Transportation Network Model based on Time-indexed (TNMT) formulation
- Provides highly structured optimal solutions

Transportation Network Model based on Time-indexed (TNMT) Formulation

- Time window discretization and indexing



- Binary decision variables

$$u_{ji} = \begin{cases} 1 & \text{if one unit message } j \text{ is delivered in interval } i \\ 0 & \text{otherwise} \end{cases}$$

- Objective function: Approximate weighted completion time [1]

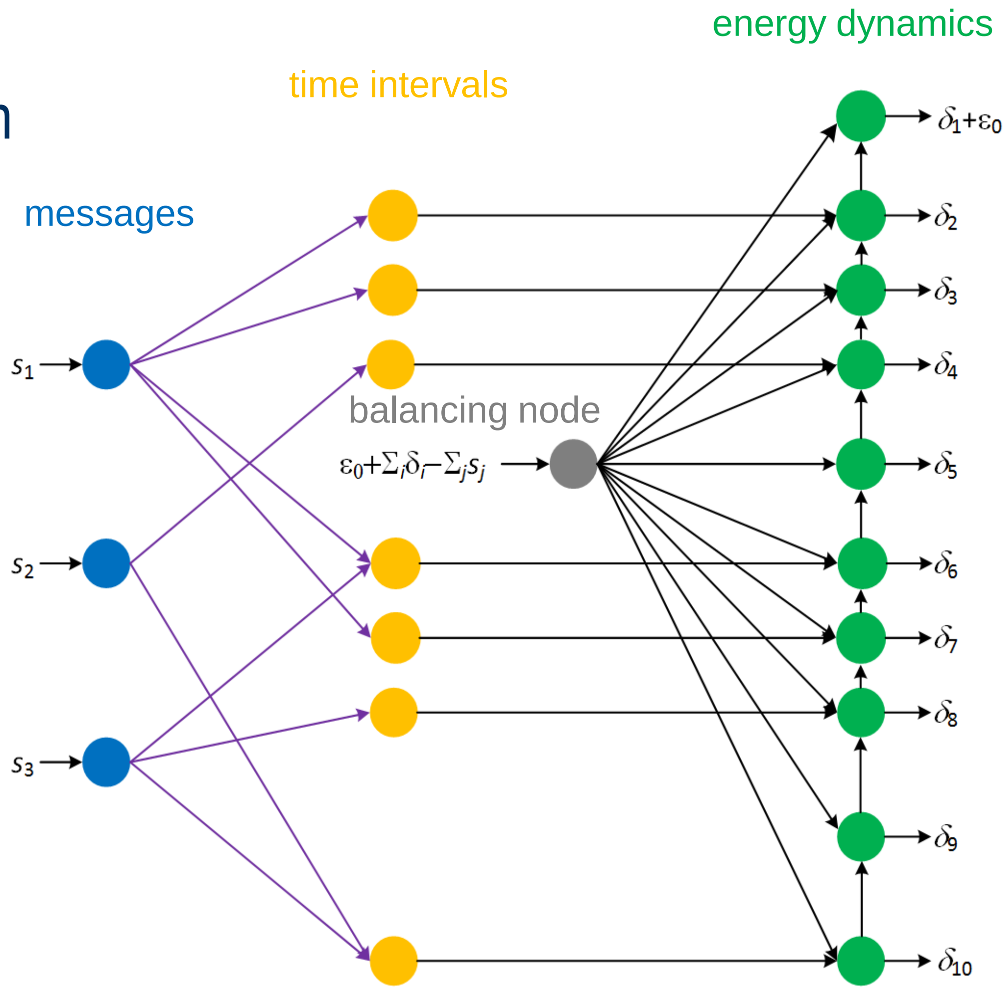
$$\sum_j \left(\sum_i \frac{w_j}{s_j} \left(t_i + \frac{1}{2} \right) u_{ji} + \cancel{\frac{w_j}{2} s_j} \right)$$

t_i = start time of interval i
w_j = priority weight
s_j = message size

[1] M. X. Goemans, et al., "Single Machine Scheduling with Release Dates," *SIAM J. Discrete Math.*, 2002.

Transportation Network Representation

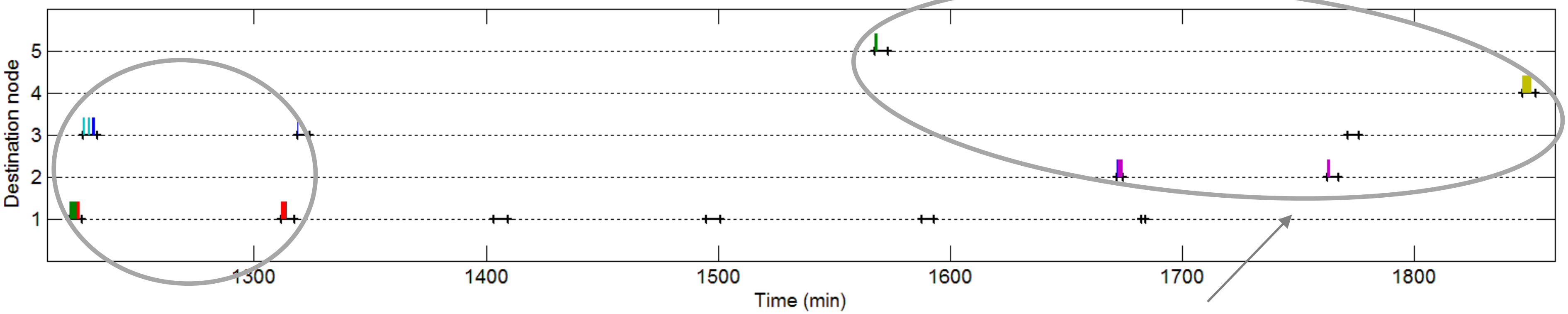
- We make the approximation such that the resulting problem is a network problem
- Constraints
  - No more than one job per time interval
  - Job completion
  - Energy dynamics
- Network flow property
  - Feasibility (supply=demand)
  - Integrality (integer supplies and demands)
  - Efficient network algorithms exist (Hungarian algorithm, Auction algorithm)



Scheduling Strategy Comparison

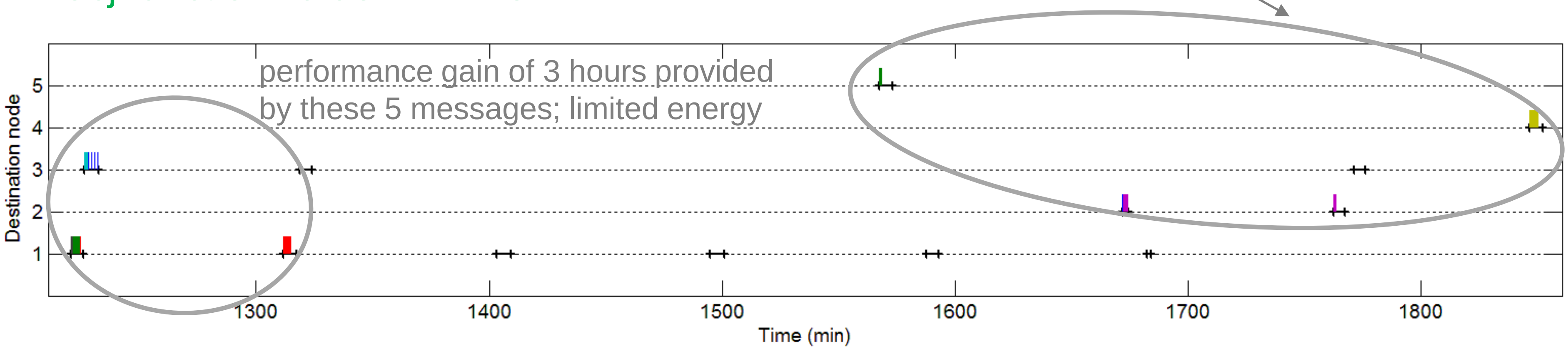
TNMT outperforms Greedy by 3 hours in total delivery time.

obj function value = 1732693 **Greedy**



same C_j; not limited by energy

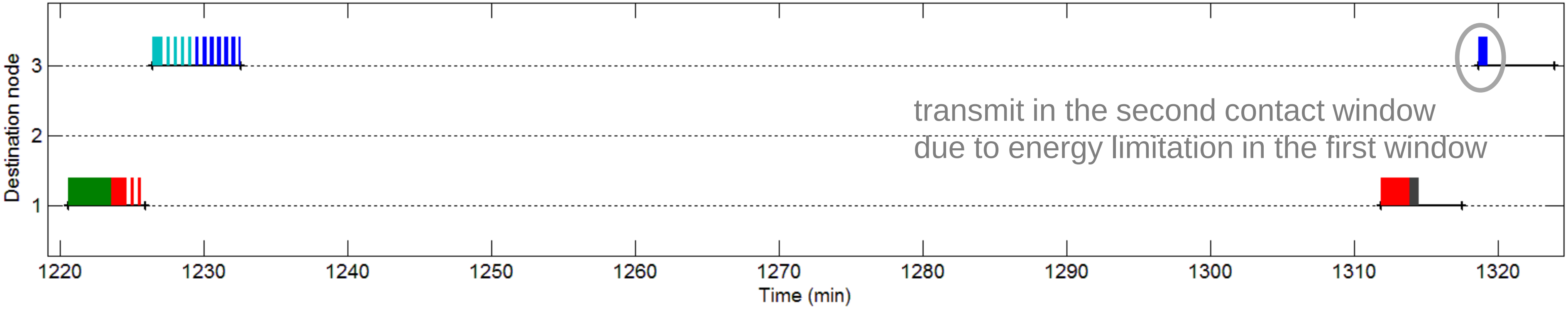
obj function value = 1711572 **TNMT**



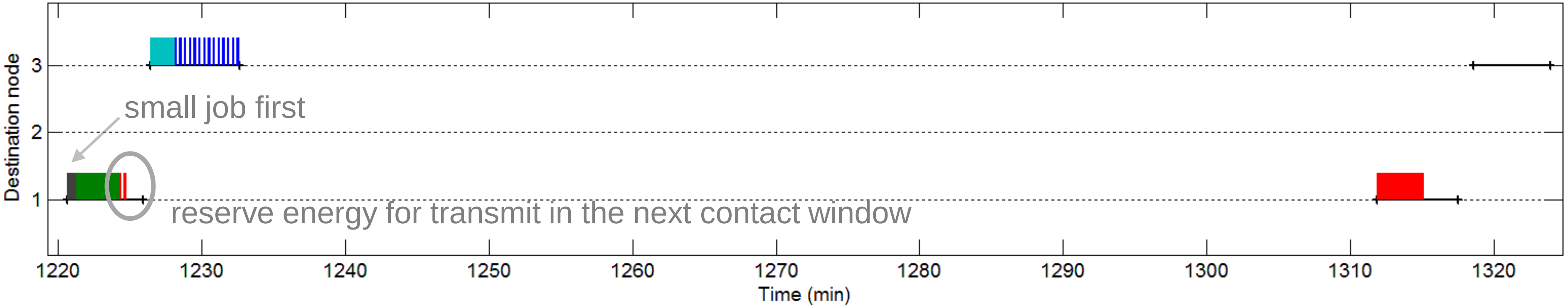
Scheduling Strategy Comparison

TNMT outperforms Greedy by 3 hours in total delivery time.

Greedy



TNMT



Advantages of TNMT

- Optimizes approximate weighted completion time
- Includes lookahead for better planning especially when limited by energy
- Balances message priorities, sizes and satellite energy over contact time windows
- The network model allows efficient computation
- Integer solutions guaranteed with integer data

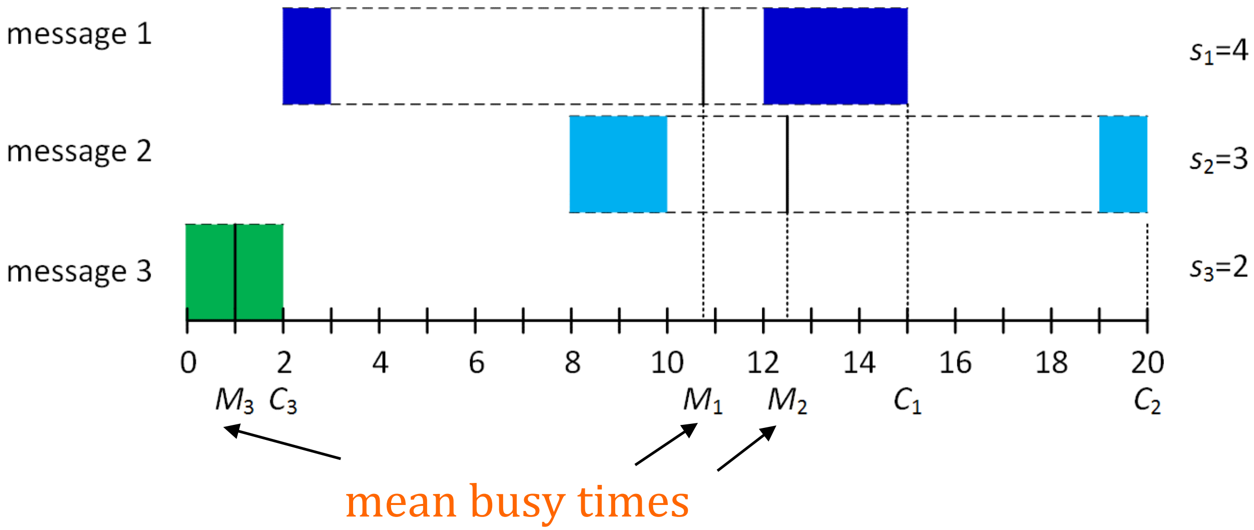
Summary and Future Work

- Summary
  - Developed TNMT to find high quality solutions quickly
  - Used transportation network model with special properties
  - TNMT gives better solution than greedy when energy is limited

- Ongoing & Future Work
  - Online algorithm for dynamically updating message delivery schedules with new message arrivals
  - Priority adjustment scheme to guarantee message delivery
  - Address congestion in multiple nanosatellite scheduling

Backup Slides

Mean Busy Time



$$M_1 + \frac{s_1}{2} = C_1$$
$$M_2 + \frac{s_2}{2} < C_2$$
$$M_3 + \frac{s_3}{2} < C_3$$

LEMMA [1]

For any job j, we have $M_j + \frac{1}{2}s_j \le C_j$, with equality if and only if job j is not preempted.

[1] M. X. Goemans, et al., "Single Machine Scheduling with Release Dates," *SIAM J. Discrete Math.*, 2002.

Numerical Example

- Scheduling Strategies
  - TNMT (Transportation Network Model based on Time-indexing)
  - Greedy (Highest priority first)

Message	Destination	Size (sec)	Priority Weight		Greedy Completion Time (sec)	TMNT Completion Time (sec)
1	2	30	3		100335	100335
2	**1**	**180**	**3**		**73416**	**73456**
3	**1**	**220**	**2**		**78831**	**78910**
4	**3**	**100**	**3**		**73756**	**73687**
5	2	230	1		105827	105827
6	4	230	2		111039	111039
7	**1**	**40**	**1**		**78871**	**73276**
8	**3**	**140**	**3**		**79156**	**73957**
9	5	90	2		94133	94133