THE OHIO STATE UNIVERSITY

ELECTROSCIENCE LABORATORY

Using Cognitive Communications to Increase the Operational Value of Collaborative Satellite Networks

Ryan Linnabary, Andrew O'Brien, Graeme E. Smith, Christopher Ball, and Joel T. Johnson

linnabary.24@osu.edu

IEEE Cognitive Communications for Aerospace Applications Workshop - Cleveland, Ohio

06/26/2019

NASA Advanced Information Systems Technology Program NNH16ZDA001N-AIST

Outline

Motivation

- NASA's future space systems will include[1]:

  Heterogeneous Networks of Autonomous Small Satellites

- Distributed missions - smaller platforms - but greater numbers
- Resource constraints
  - Low power, restricted duty cycle
  - Limited link availability/reliability, buffer size, reduced power/BW
- Instrument data volume & sensor reconfigurability will expand[2]
- But down-links are not always appropriate *(or possible)*
- Space systems will rely more on inter-satellite communication

[1] Gilbert J. Clark, Wesley Eddy, Sandra K. Johnson, David E. Brooks, and James L. Barnes, "Architecture for Cognitive Networking within NASA's Future Space Communications Infrastructure", 34th AIAA International Communications Satellite Systems Conference.

[2] A. G. Schmidt, G. Weisz, M. French, T. Flatley and C. Y. Villalpando, "SpaceCubeX: A Framework For Evaluating Hybrid Multi-Core CPU/FPGA/DSP Architectures", IEEE Aerospace Conference, 2017

- New mission science goals will depend on collaboration and autonomy
- Introduces a complex decision-making space
- Appealing solution[3]: **Cognitive Communication**
- However, it is insufficient to simply increase link capabilities
- Also necessary to improve the complex communication decision-making:
  - Deciding when to communicate
  - What information is valuable to nodes of the network
  - How to adapt local operations based on new information
- Challenges:
  - Optimizing mission science return *(remote-sensing, etc.)*
  - Increasing the effectiveness of resource-constrained systems
  - Deploying cognitive algorithms on small-satellite platforms

[3] J. Barnes et. al., "Machine Learning for Space Communications Service Management Tasks", CCAAW 2017

Approach to the Problem

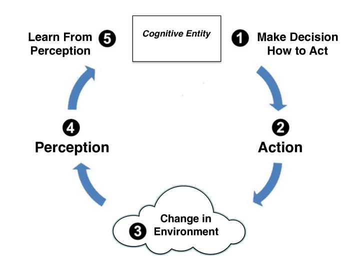
- We developed an open-source C++ space-network simulation library[4]
  - *Purpose*: Efficient simulation of satellite networks with realistic constraints *(power, sensor hardware, communications)*
  - *Key focus*: Simulate sensors that make intelligent decisions based on their own measurements and measurements provided by the network
- We performed simulations to apply cognition to collaborative small sats
- We are investigating:
  - Production of large training data-sets that capture network operation
  - Machine learning techniques to make intelligent communications decisions
  - The applicability of these methods to future cognitive space communication

[4] R. B. Linnabary, A. O'Brien, G. E. Smith, C. D. Ball, and J. T. Johnson, "Open Source Software for Simulating Collaborative Networks of Autonomous Adaptive Sensors", IGARSS, 2019

Outline

Cognition

- Selecting and carrying out actions based on both:
  1. Specific goals
  2. Perception of environment
- Learning from experiences
- Interaction with environment



Cognitive Entity

- An intelligent system that possesses perception, learning, reasoning, and decision making capabilities[a]

---

[a] G. E. Smith, A. E. Mitchell, C. D. Ball, A. O'Brien, and J. T. Johnson, "Fully Adaptive Remote Sensing Observing System Simulation Experiments", IGARSS, 2018

Communications which are *operationally dependent* on cognition

Most Existing Work *(Low Level)*

- Cognitive optimization of software-defined radio parameters[5]
- Cognitive satellite digital beamforming[6]
- Bandwidth, Power, Frequency, Modulation

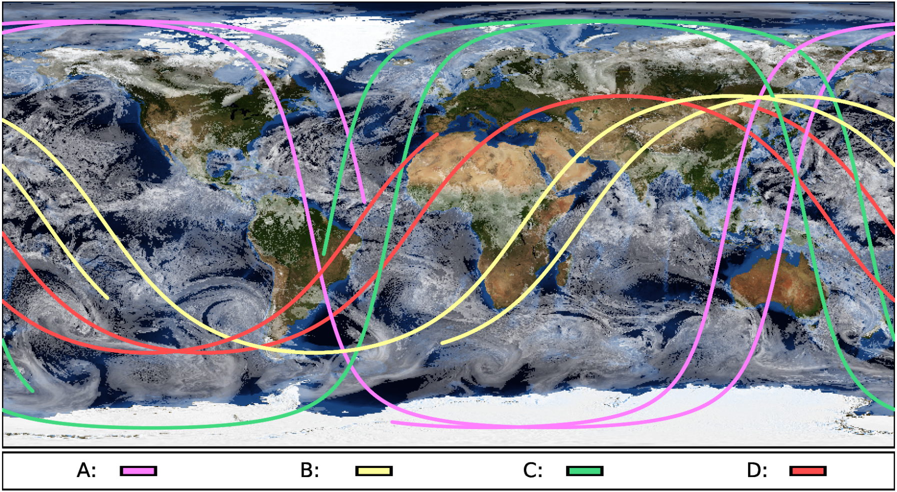
A New Approach *(High Level)*

- Intelligent routing of information within autonomous network[a]
- Data Contents, Routing, Latency, Sensor Parameters

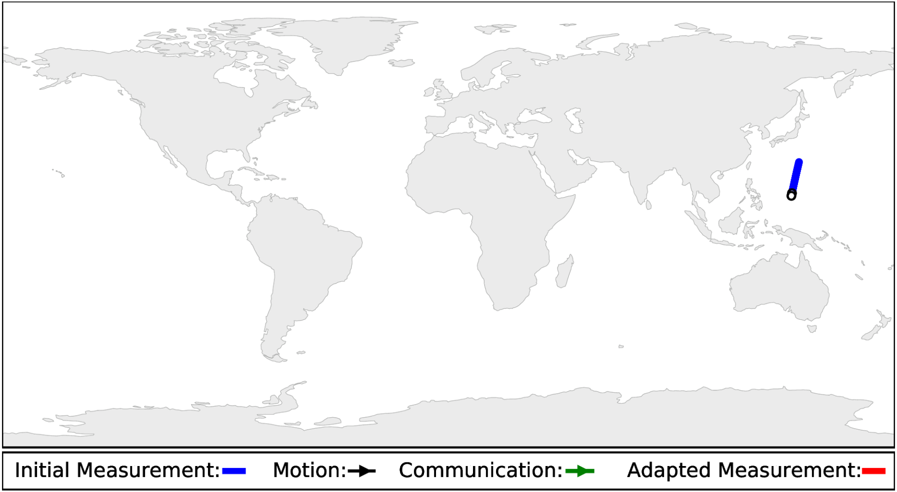
[5] P. V. R. Ferreira, et al., "Multi-Objective Reinforcement Learning-Based Deep Neural Networks for Cognitive Space Communications" CCAAW, 2017

[6] Wenhao Xiong, J. Lu, X. Tian, G. Chen, K. Pham and E. Blasch, "Cognitive Radio Testbed for Digital Beamforming of Satellite Communication" CCAAW, 2017

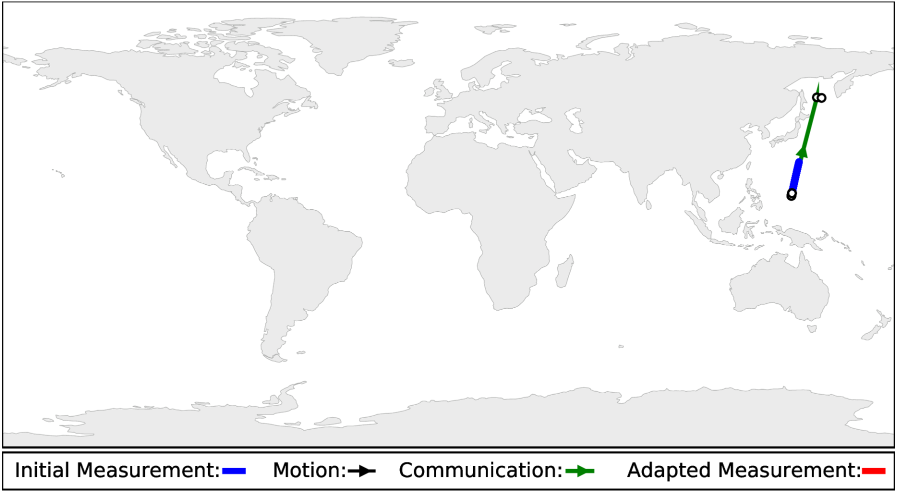
Four satellites in separate orbit planes, moving in opposite directions



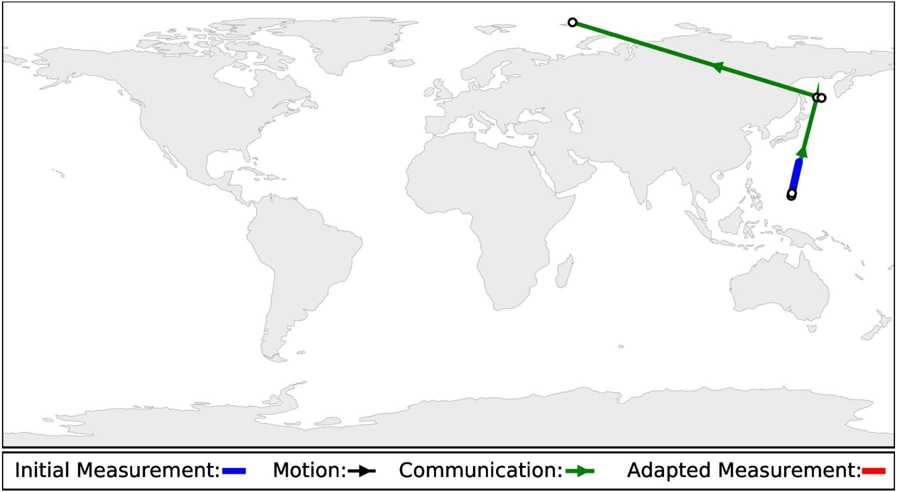
Satellite A senses atmospheric data and performs *internal processing*



Satellite A forwards a new packet to **Satellite B**

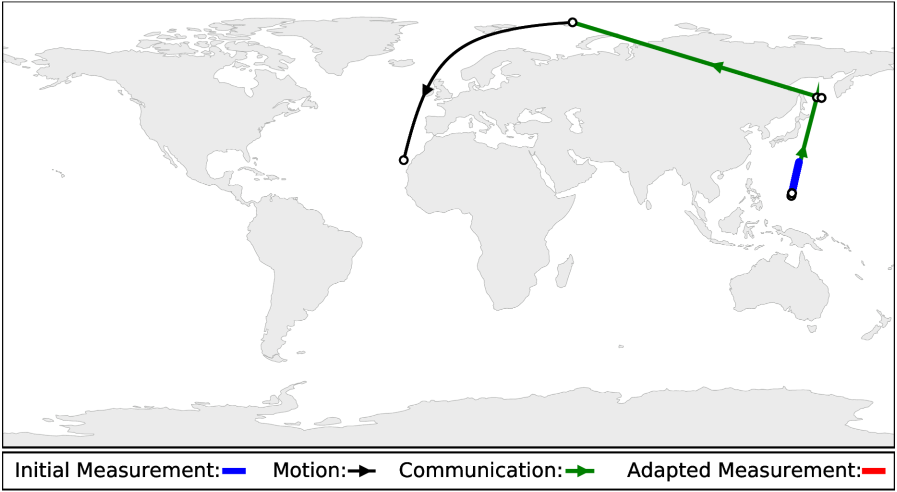


Satellite B forwards the packet to **Satellite C**

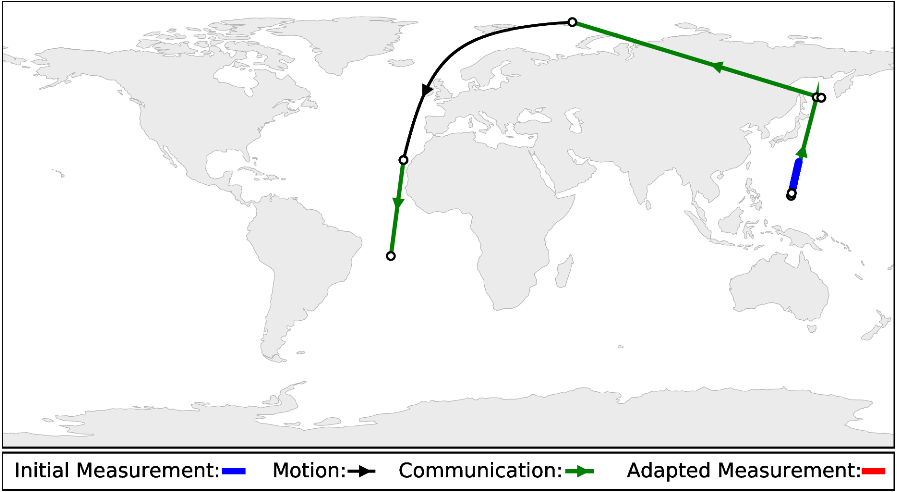


Collaborative Route

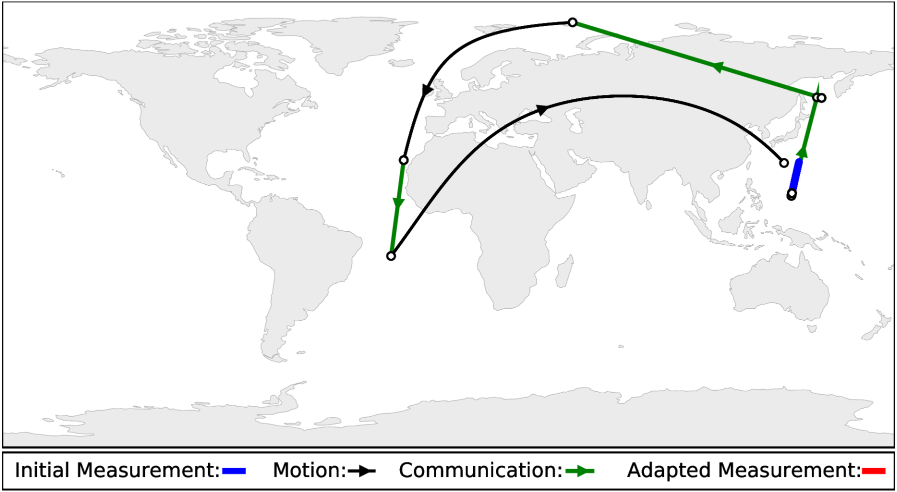
Satellite C holds the packet, *anticipating* an opportunity to transmit



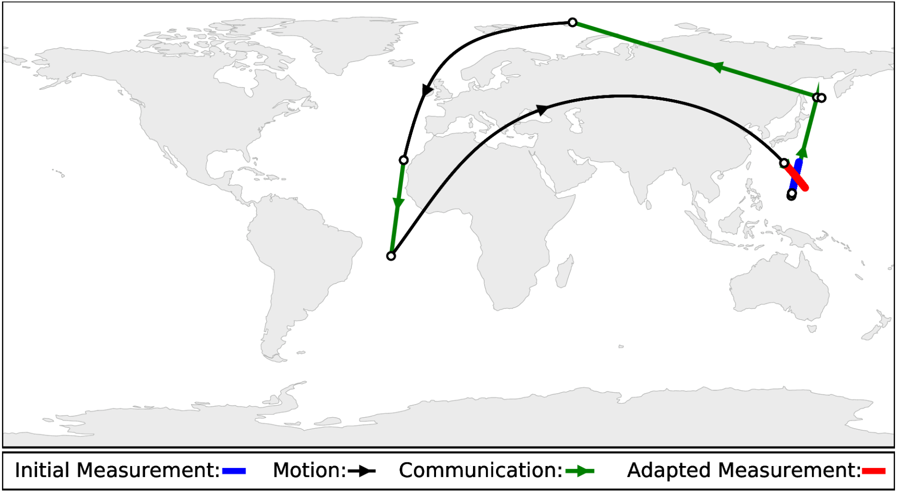
Satellite C forwards the packet to **Satellite D**



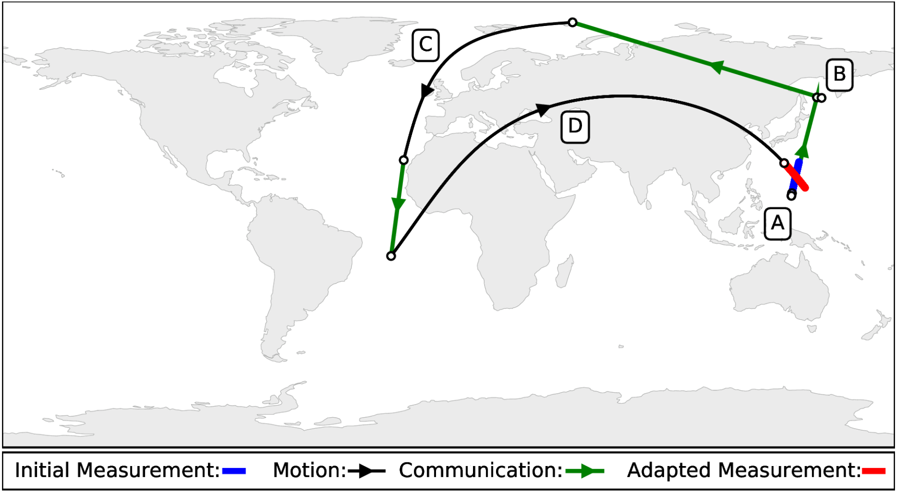
Satellite D processes the packet and adapts its hardware or behavior



Satellite D performs a *follow-up* measurement

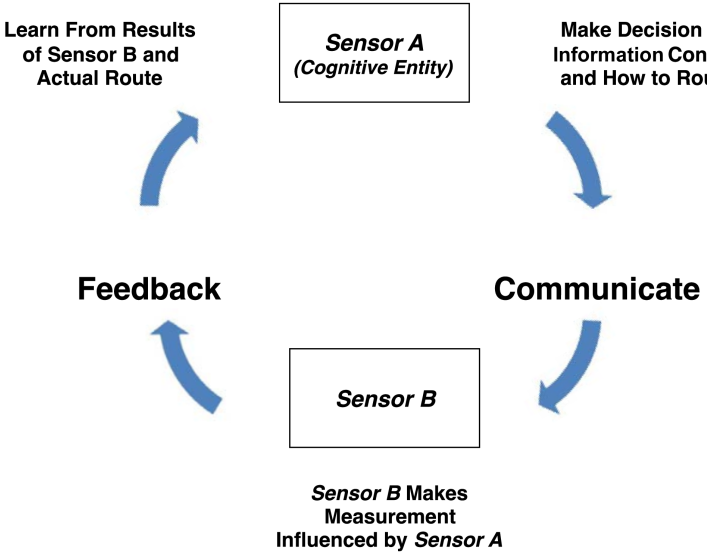


A long-term illustration of the collaborative algorithm



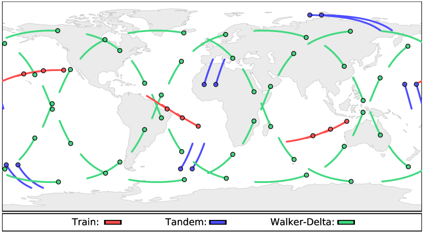
Cognitive Network (Collaboration)

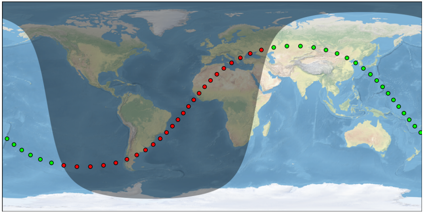
The Ohio State University

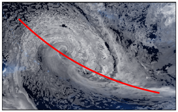


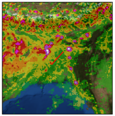
- COLLABORATE is a software toolset under development for producing training data and implement ML algorithms
  - A C++ development library for observing system simulations
  - A Python package for simulation-data processing (vis./analysis)
- Simulates collaborative networks of satellites
- Focuses on the high-level communication decision space
- Employs network data structures (trees/graphs) to execute predictive route-finding algorithms for efficient communications
- Offers many unique features valuable to future observing system simulation experiments
- Project published to a Git repository, licensed LGPLv3.0.[7]

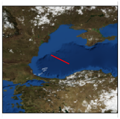
[7] https://www.github.com/ryananan/collaborate/

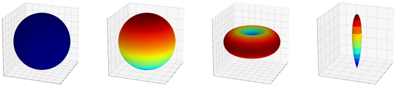

Train:
Tandem:
Walker-Delta:

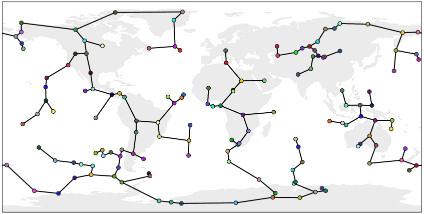


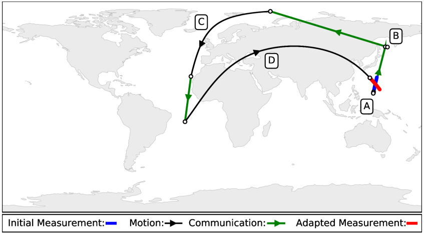











C
B
D
A
Initial Measurement:
Motion:
Communication:
Adapted Measurement:

Advantages

- Latency-tolerant *(data rate & bandwidth)*
- Predictive *(route participants can retain data)*

Considerations

- Antenna orientation
- Relative velocity *(Doppler)*
- Limited buffer size, channel interruptions

Assumptions

- Line-of-sight connections with power threshold
- Free-space links
- Established among compatible devices
- No spectrum conflicts *(yet)*

Example – Simulation Setup

Two Main Constellations

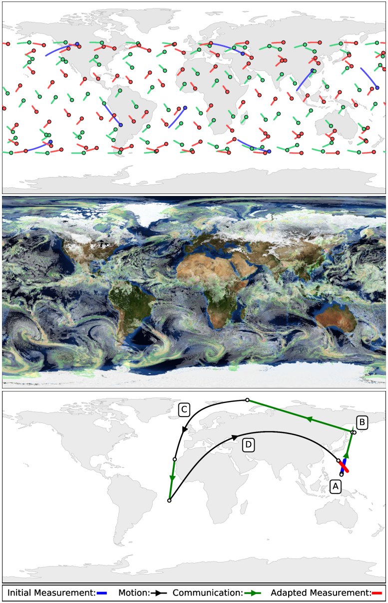
- Cloud Radar
  - Has a *threshold parameter*
- Precipitation Sensor

Atmospheric Data

- Total cloud optical thickness
- Total precipitation

Exploit Data Correlation

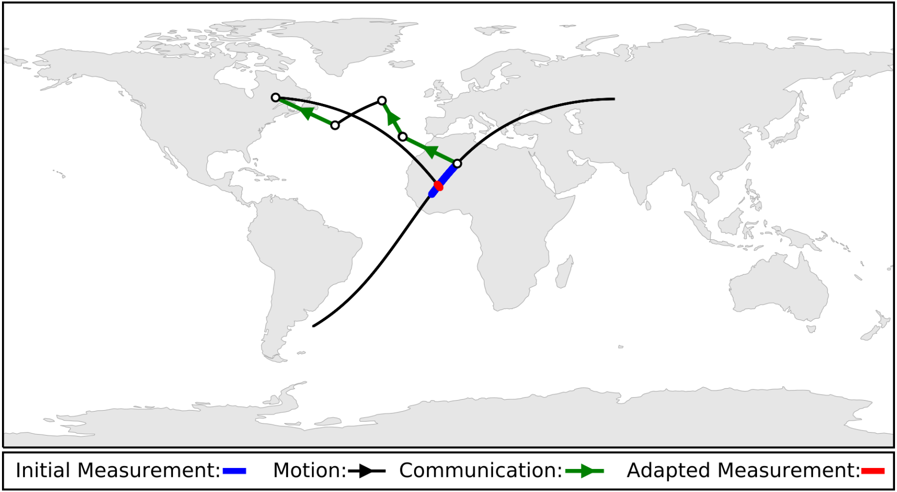
- Goal: maximize precipitation measurements
- Measure cloud thickness at maximum duty-cycle
- Cue precipitation sensors to target regions with high clouds



Example – Simulation Steps

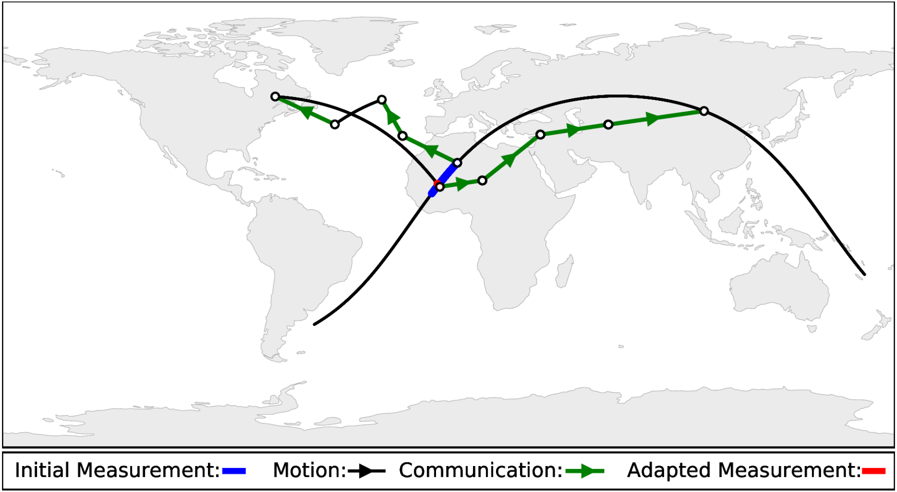
1. A cloud radar **"A"** measures cloud optical thickness
2. Cloud thickness exceeds established threshold
3. **"A"** predicts precipitation sensor **"B"** will visit at time τ
4. **"A"** forms packet containing *route*, *environment*, and *instructions*
5. Packet is forwarded through network before time τ
6. **"B"** takes follow-up measurement
7. **"B"** forms packet containing *route*, *environment*, and *instructions*
8. Packet is forwarded through network ASAP
9. **"A"** modifies internal threshold based on success criteria
10. Cycle is repeated

Feed-Forward Route



1. A cloud radar **"A"** measures cloud optical thickness
2. Cloud thickness exceeds established threshold
3. **"A"** predicts precipitation sensor **"B"** will visit at time τ
4. **"A"** forms packet containing *route*, *environment*, and *instructions*
5. Packet is forwarded through network to **"B"** before time τ
6. **"B"** takes follow-up measurement
7. **"B"** forms packet containing *route*, *environment*, and *instructions*
8. Packet is forwarded through network to **"A"** ASAP
9. **"A"** modifies internal threshold based on success criteria
10. Cycle is repeated

Feedback Route

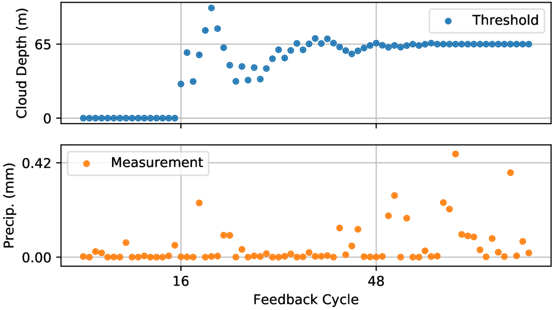


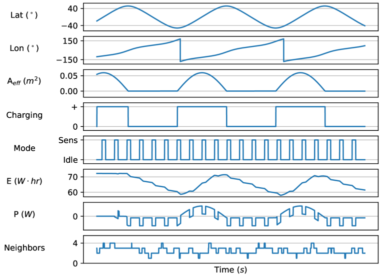
Example – Simulation Steps

1. A cloud radar **"A"** measures cloud optical thickness
2. Cloud thickness exceeds established threshold
3. **"A"** predicts precipitation sensor **"B"** will visit at time τ
4. **"A"** forms packet containing *route*, *environment*, and *instructions*
5. Packet is forwarded through network to **"B"** before time τ
6. **"B"** takes follow-up measurement
7. **"B"** forms packet containing *route*, *environment*, and *instructions*
8. Packet is forwarded through network to **"A"** ASAP
9. **"A"** modifies internal threshold based on success criteria
10. Cycle is repeated

Simulation steps are repeated for several cycles until the sensors in the network have adapted based on feedback from one another

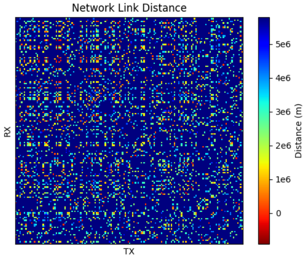
- Cycles 1-16: Measure precipitation anywhere
- Cycles 16-48: Regression
- Cycles 48-∞: Measure precipitation where clouds $\geq$ 65 meters





Simulation Data

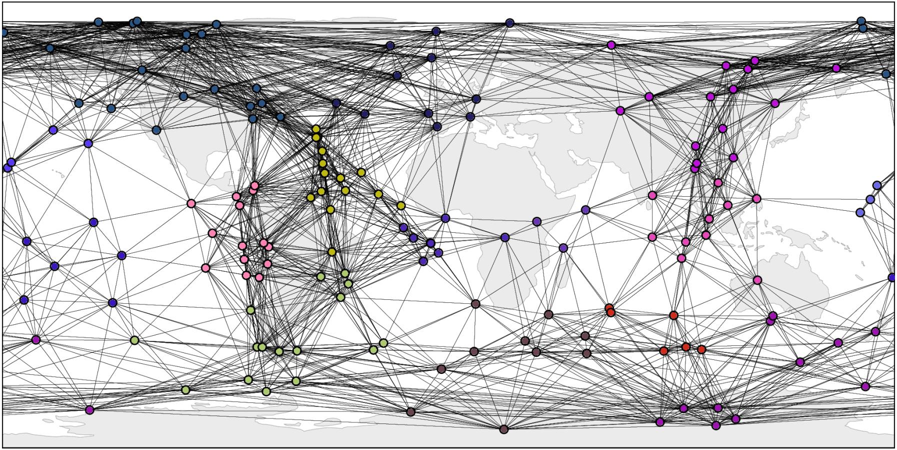
- NetCDF format
- Satellite parameters
- Communication channel data
- Network structures
- Sensor measurements



Machine-Learning Model

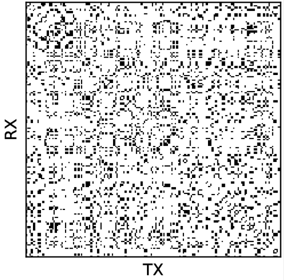
- Python
- Scikit-Learn
- *Input*: weighted adjacency matrices
- Spectral clustering (k-means)

All line-of-sight connections in network *(potential links)*

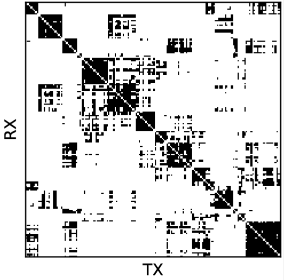


- Classify satellites based on *proximity*
- Extendable to other satellite parameters in the simulation data
- May inform beam direction, channel decisions, TX power, etc.

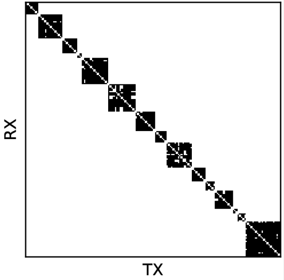
Lines-of-Sight



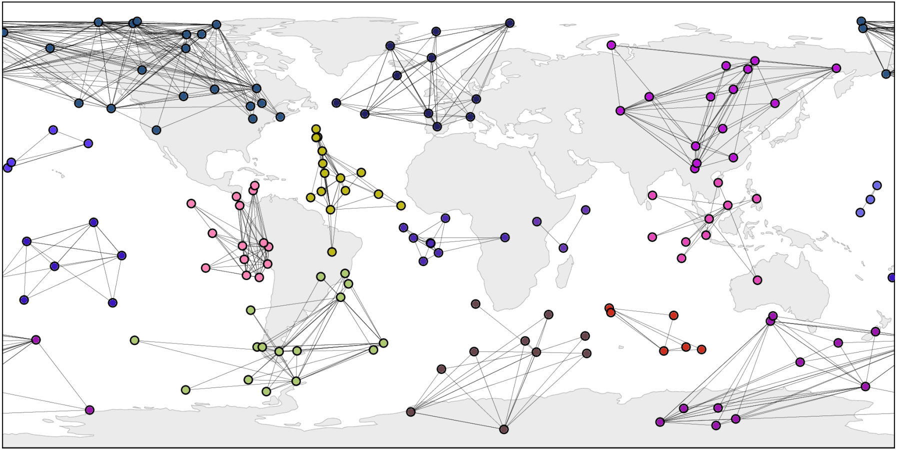
Sorted Network



Clusters



Isolated satellite network clusters



Conclusion and Future Work

- Future small-satellites will carry adaptive instruments
- Parameters will be intelligently re-configured
- *Primary purpose collaborative communication* among small satellites:

Achieve system-level adaptivity with future instruments

- We introduced our current work:
  - Applying cognitive communications to information flow in a collaborative small-satellite network
  - Using COLLABORATE to investigate cognition as one means of overcoming the complex decision space for such small satellites
- Future goal:
  - Simulate more advanced ML algorithms (employ Neural Network)
  - Add valuable algorithms identified in Python to the C++ routines