# Features To Develop

* Relax Constraints on Analysis
  + Fairness (Other processes can become leader)
    - Seems really heard. Involves coming up with a way of capturing the remaining portion of the state (other groups) through an actual value, or through a state distribution.
    - It may be an issue finding or using Chi-Square tables for large numbers of states since the DoF grows exponentially.
  + Livelock
* Bounding growth limits (How the algorithm behaves with 100’s of processes) and identifying any interesting algorithm behaviors with that many processes.
* Is there a bound argument that can be made for more complex system descriptions? (ie, IGT will be between these two bounds, based on these simple models) Is there a strict > relationship for some configurations?
* More complex loss profiles: more complicated message loss schemes based on other protocols? One idea from previous research was that the communication protocol also needed the memoryless property so it doesn’t get “clogged” (a queueing issue)
* More complex loss profiles: Message loss that isn’t uniform throughout the system.
* Hardening techniques for group management/General algorithms?
* Developing a general technique or framework for designing distributed algorithms?
* Journal Related Tasks
  + More about the approach and applications for distributed system design
  + Using these techniques to anticipate failure.
* More motivation… find other applications outside of smart grid to use as toy problems?
* Other algorithms? (Homogenous coin flipping, bully?)

# Applications

* Using ECN to anticipate failure and harden the distributed algorithms.
* Other Non-ECN anticipation techiques?
* Producing omission rates accurately – minimize error with w/ limited rolling observations?
* Apply the anticipation to protect other algorithms – can they be changed transparently?
* Using the anticipation to apply other techniques to maintain a level of service for other applications.
* Using the large results to put a bound on group size.
* Use the result to test performance of algorithms that rely on group management like LB
* Journal Related Tasks
  + Clarify what the stacked bar graph means. Other practical applications beyond just providing insight for readers?
  + Using the model to bound group size.

# Writing Tasks For Journal

* More Motivation for this approach.
* More references to LB and UNI
* Develop reasoning for changes/why the changes are reasonable
* Crispen paper
* Move the algorithm text to the appendix and make it a state machine?