Design of a Fault-Tolerant Satellite Cluster Link Establishment Protocol

Stephen Horan, Praveen Thonour, & Krishna Mohan Chaganti

Klipsch School of Electrical and Computer Engineering

This material is based upon work supported by the National Aeronautics and Space Administration under Award No. NAG5-13189.

Topics

- Goals and Assumptions
- Design Methodology
- Phase I Results
- Phase II Improvements
- Phase II Lessons Learned
- Next Steps
- Conclusions

Goals and Assumptions

- Develop a protocol to address link establishment for satellite clusters
  - Assume 10’s to 100’s of nodes are possible in the network
  - Assume not all nodes are mutually visible at all times
  - Assume channel errors are possible
  - Assume satellites may fail and recover

Goals and Assumptions

- Operational assumptions
  - Satellite cluster access from the ground will be limited to well-known access points and trusted users
  - Many nodes and access points can be pre-loaded before launch.
  - Algorithm will autostart to determine node configuration and connections.
  - Wish to minimize the amount of overhead in transmitting path information.

Design Methodology

- Use the *LabVIEW* toolkit *State Diagram* to design the protocol software
- Designing a modular state diagram to realize the algorithm
- After testing the basic algorithm in *LabVIEW*, the algorithm is converted to ANSI C reference code for detailed testing.
- Use a multi-node laboratory network for algorithm testing over the lab LAN.

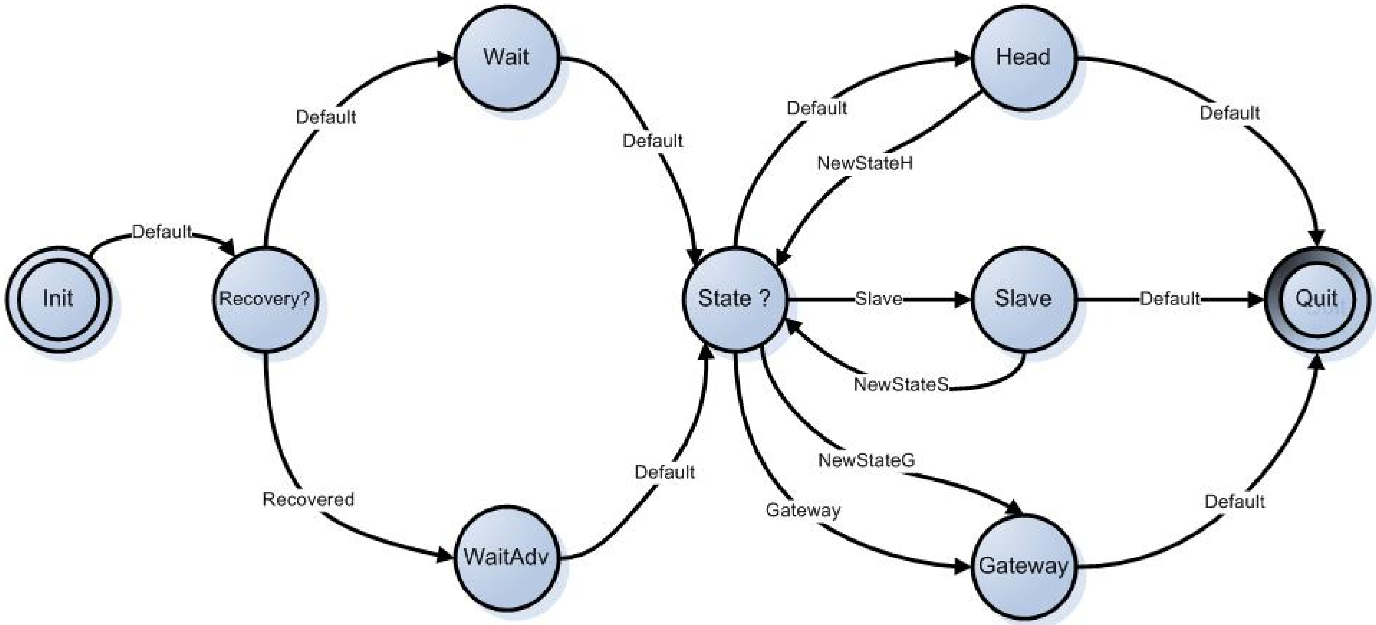
Phase I Results

- Demonstrated
  - Basic control algorithm was developed and worked as intended
  - Developed an ANSI C reference version of the protocol
- Learned
  - Need some form of FEC on the management messages when the channel BER is worse than 10^{-3}
  - Needed to improve message passing to avoid collisions
  - Needed additional management functions to make the protocol more versatile

Phase II Improvements

- The second phase made the following improvements based on the Phase I results:
  - Utilize multicast messaging to improve channel use efficiency,
  - Provide a method to allow new members to join the cluster,
  - Provide a method to partition the cluster into smaller partitions of single-hop neighbors, and
  - Provide a means for gateways between partitions.

Phase II Improvements



- Modified Phase I state diagram for new functions
  - Added a Gateway state
  - Synchronized management messages with token

Phase II Improvements

- Phase II Modifications (cont.)
  - Added a periodic advertisement message to coordinate new member discovery
  - Added multicast message transmission for management messages
  - Added means for automatic partitioning of total cluster to smaller, single-hop partitions

Phase II Lessons Learned

- The current version of the protocol has a number of parameters that specify its operations: token parameters, advertisement messages, etc.
  - These parameters need to be well tuned for the operational hardware configuration to have the best operational results
  - The operating system clocking can have a major effect on the timing of the parameters

Phase II Lessons Learned

- Needed to create dual multicast channels to simulate the partitioning effects of distance on the single LAN. This adds overhead to the protocol that can be removed if an actual radio link is used.

Next Steps

- Next major step is to try to run the protocol over a radio link
  - Looking to use Freescsale radios to try out the protocol on a low-power radio evaluation unit
  - Test operations and performance in this environment with the reference ANSI C code.

Conclusions

- The Phase II development was able to add features such as multicasting, automatic partitioning, advertising, etc. to the Phase I protocol.
- Found a need to careful tuning of protocol parameters for efficient operation; some parameters sensitive to operating system conditions for timing and synchronization.