Synchronized Embedded Filesystem over ANT

Ethan Chen
Wenchuan Weng

Problem

- Wireless sensor network nodes may need shared state or shared data.
- Wireless sensor network nodes may not have the global picture.
- Synchronization over distributed network is tricky.

Issues of Synchronized Filesystem

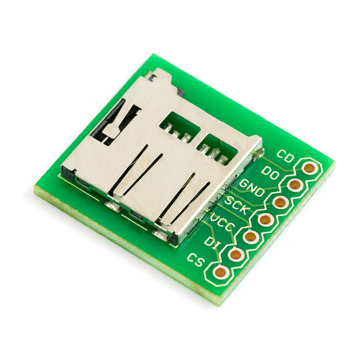
- Resolving conflicts
  - clock synchronizaion
- Choosing the right level of abstraction
  - Block level
  - File system level: directory tree, file
- Efficiency
  - Obvious solution: copy everything over
  - Better: compress everything and copy over
  - only send over differences

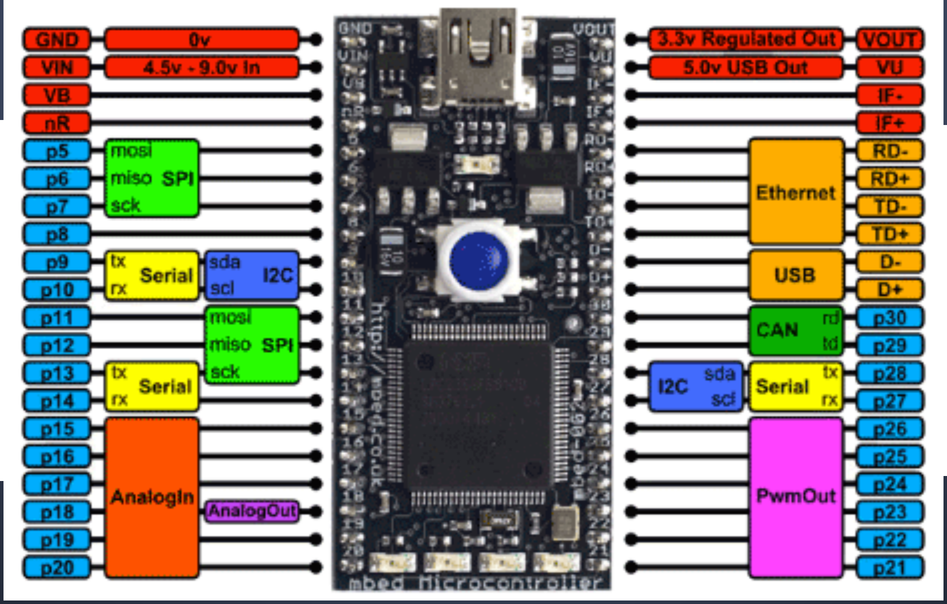
Our Proposed Solution

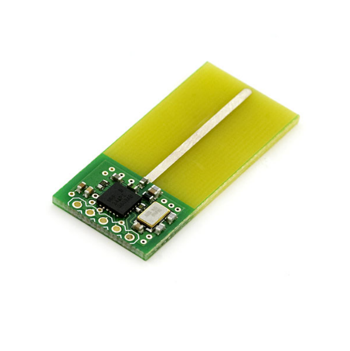
- Utilize a global master in charge of directing all node synchronization
- Block level synchronization utilizing hash to detect change
- Master acts as global lock
- Modification of blocks on a node requires node to notify master
- Master broadcasts updates

Hardware

- Node: mbed (NXP LPC1768)
- Storage: SD card
- Networking: nRF24AP1



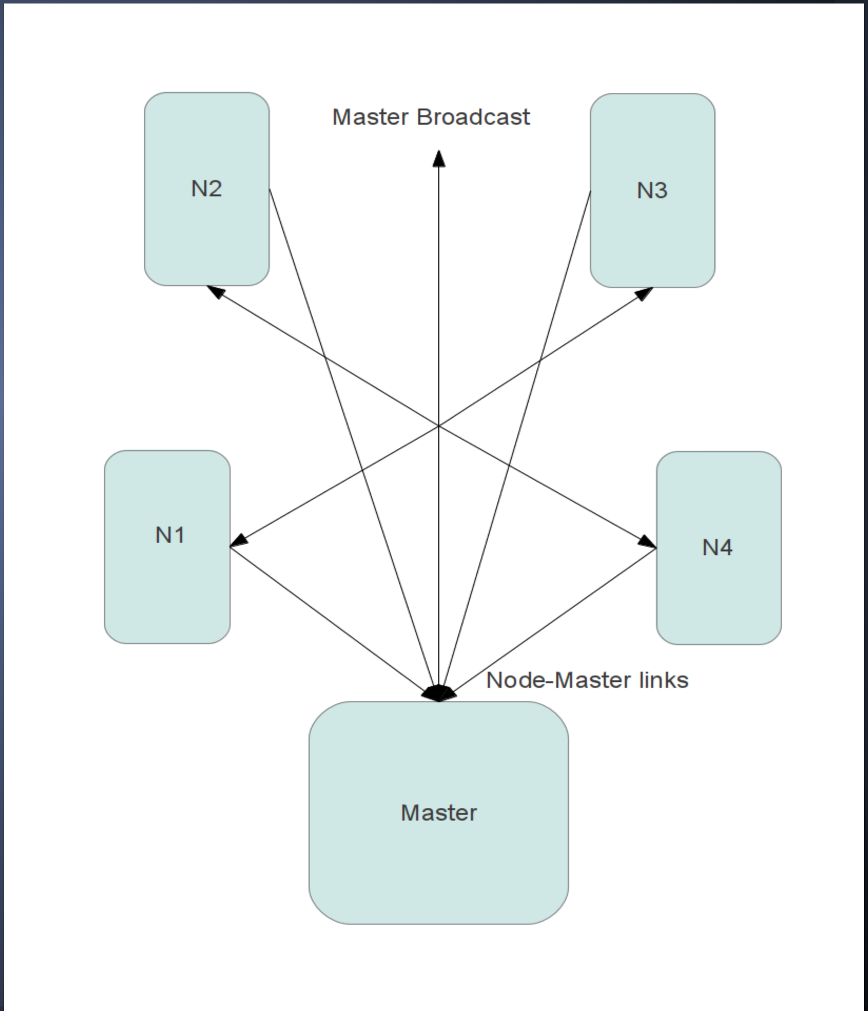




Network Architecture

Spoke-Hub based architecture:

- Master broadcasts presence
- All nodes listen on broadcast
- Nodes connect direct to master for duplex link
- Synchronization over slow link is difficult



Network Architecture (continued)

Network packet protocol:

- Custom format ontop of ANT.
- Links are constantly "live".
- Dummy packets used for synchronization, discovery.
- Reliable transport utilizing ANT ACK.
- Data transfer utilizing ANT Burst Data mode.

Synchronization Primitives

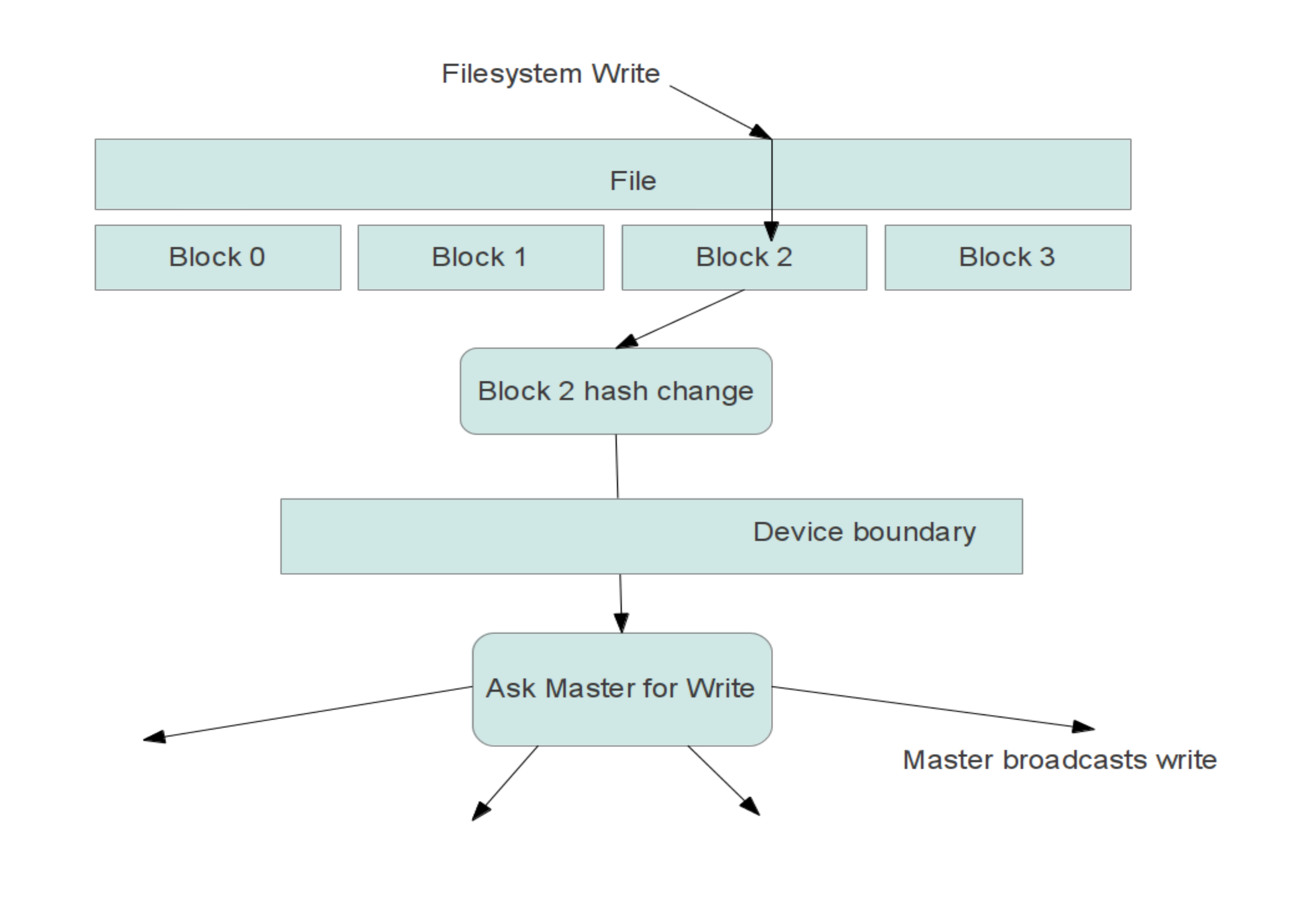
Message types:

- TIME (M -> S)
  master broadcasts timestamp
- PUSH (M <--> S)
  slave request changes to be made, master broadcasts
- STATUS (M -> S)
  master broadcasts MD4
- PULL (M <- S)
  slave discover differences with master, pull block contents
- UPDATE (M -> S)
  master send requested block contents to slave

Filesystem Architecture

- Slaves send change requests to master along with timestamps with PUSH primitive
- Master resolve conflicts and broadcasts updates to slave nodes
- Master periodically calculates a strong 128-bit MD4 checksum of each block and broadcasts checksums with STATUS message
- Slave check its blocks' MD4 checksums with master's copy, request specific block to be updated when there's difference.

Filesystem Architecture (continued)



Shortcomings

- nRF24AP1 chip used is limited to 4800 bps due to its host interface.
- ANT protocol incurs high overhead, especially when implementing another protocol on top of it.
- Readily available drivers and documentation for nRF24AP1 is lacking.
- Master node becomes system bottleneck.
- Slow network makes resolving conflicts more difficult.

Shortcomings (continued)

- ANT data transfer channel is one-way, requiring 2 channels for bidirectional communication.
- ANT data transfer channel is packet stream based and not well suited for random data transfer.
- Overhead of constantly handling packet data cuts into power, performance.

Conclusion

- More flexible network module required.
  - Need good point to point performance.
- Need to leverage more off-the-shelf software.
  - Especially network driver, protocol
- Low data rate makes synchronization more conflict-prone.
- Centralized synchronization is slow, but ok in this circumstance due to low data rate.

References

- ELF - Efficient Log Structured File System for Micro Sensor Nodes
- LUSTER - Wireless Sensor Network for Environmental Research
- Energy-Efficiency and Storage Flexibility in the Blue File System
- The Google File System
- Improved File Synchronization Techniques for Maintaining Large Replication Collections over Slow Networks
- Efficient Algorithms for Sorting and Synchronization