Runtime Reprogramming on TinyOS
Sans Deluge Modifications

Mat Kelly
December 14, 2012
CS895 – Adventures in Sensor Networks
Old Dominion University
Goal/Task List

- Cease current execution
- Change to reprogrammable state
- Accept new program, install on a volume
- Execute new program
1. Cease Current Execution

- Showed through “Poisoning” demo earlier in semester
  - Waited for packet with flag
  - Killed timers on reception
  - Didn’t actually stop execution, just timers

- Better accomplished through TOS NetProg interface, *e.g.*, `NetProg.reboot`
2. Change Mote to Reprogrammable State

- In DelugeP.nc, add:

  ```
  command void forceMoteToRecvState(){
    state = S_RECV
    request();
  }
  ```

- Then call it from my program
2. Change Mote to Reprogrammable State

- **Problem**: Deluge is not an interface!
  - Add interface spec to Deluge
    ```java
    interface DelugeInterface {
        command void forceMoteToRecvState();
    }
    ```
  - Add wiring:
    ```java
    components Deluge;
    ...
    App.Deluge -> Deluge\n    ...
    Uses interface DelugeInterface
    ```
SUCCESS!
...Not Quite

• These steps required Deluge modifications

• Alternatives:
  1. Bring Deluge code into my app
  2. Extract the necessary functionality from Deluge to reboot into state
Digging into the Solution

- Proposed Deluge change:

```c
command void forceMoteToRecvState()
{
    state = S_RECV
    request();
}
```
Digging into the Solution

• Proposed Deluge change:

```c
command void forceMoteToRecvState()
{
    state = S_RECV
    request();
}
```
Digging into the Solution

• State:
  – enum/global var in DelugeP, inaccessible

• TOSBoot, a higher level of abstraction than Deluge
  – Tight coupling with Deluge
  – Candidate in extracting gesture code
  – Candidate in writing to flash, reloading on boot
  – Changing tosboot direction $\rightarrow$ changing deluge
Extracting Gesture Source

• Would reload Golden Image w/o modifying tosboot or Deluge

• In tosboot:

```c
startupSequence() {
  ...
  if (gestureCount > Tosboot Gesture Max Cnt) {
    ...
    programmableImage(GoldenImgAddr)
    call Hardware.reboot()
  }
}```
ProgramImage()

• Determines size of input array
• Checks that writes are allowed
• Calculates space needed, further checks
• Writes data to flash
ProgramImage()

- Determines size of input array
- Checks that writes are allowed
- Calculates space needed, further checks
- Writes data to flash

RELIES ON MANY DELUGE/TOSBOOT VARS!
The Dirty Approach

• Set the flash bit
• Reboot mote programmatically
• Cross fingers that it returns to the right state

• Key line from DelugeP:

```c
IFlash.write(
    TCAST(         
    uint8_t*COUNT(sizeof(bootArgs)),
    TOSBOOT_ARGS_ADDR),
    &bootArgs,
    sizeof(bootArgs));
```

• Requires size of volumes in memory → Obtained through Deluge interface
Conclusion

• Without Deluge modification, a user-level program cannot reboot a mote into a programmable state
• Deluge requires a public interface for this to happen