

# LiteOS based Interactive and Reliable Software Platform for Wireless Sensor Networks

ACM Sensys Doctoral Symposium 2007

Qing Cao

Advisor: Prof. Tarek Abdelzaher

University of Illinois at Urbana-Champaign

# Trends of Wireless Sensor Networks- Observations

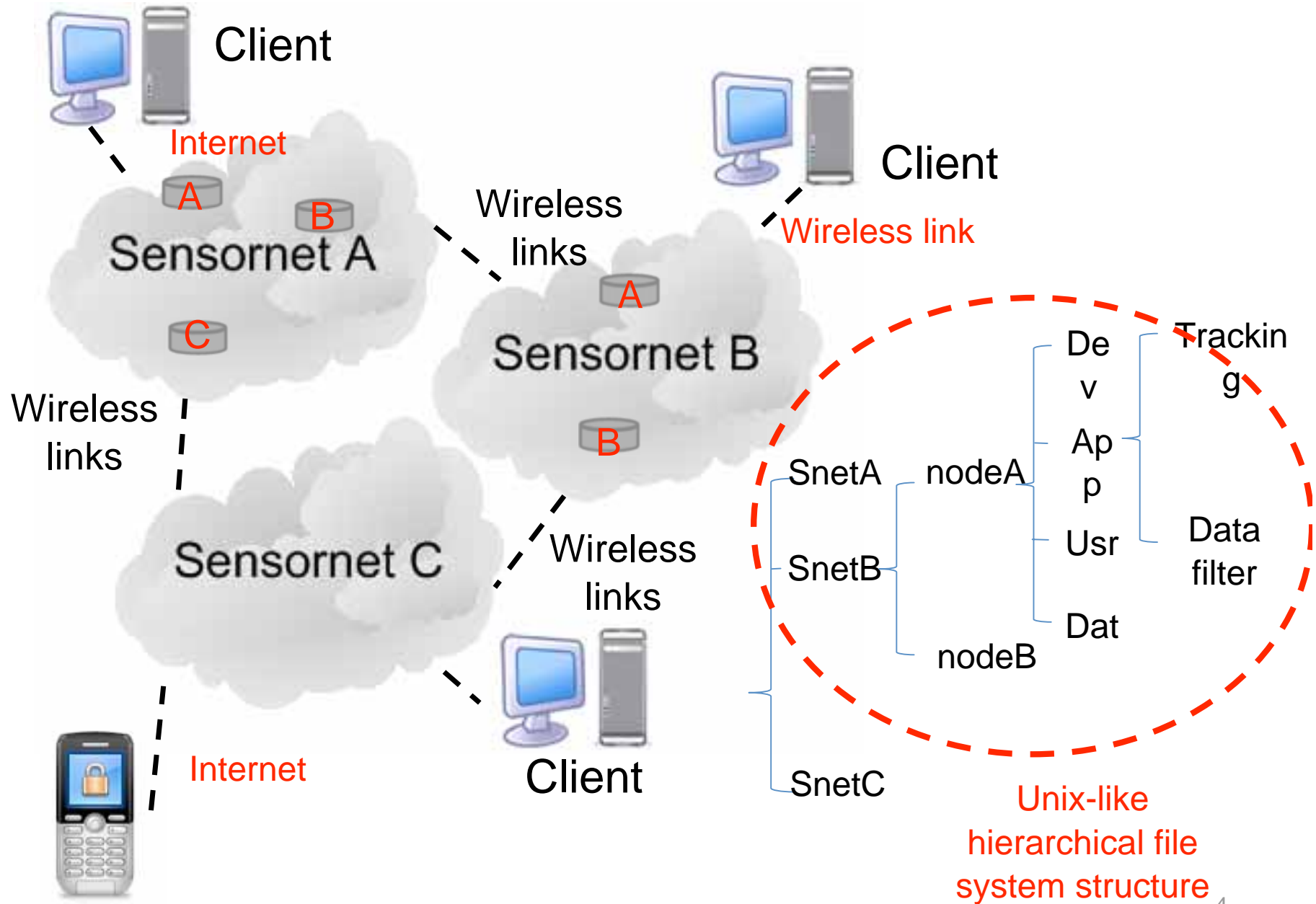
- Increasing scale and complexity of sensor network applications
  - More complicated application requirement
  - Larger deployment scale
- Increasing needs for software reliability
  - Move from toy applications to the real world
  - Mission/business critical applications
  - Emerging commercial adoptions

# Trends of Wireless Sensor Networks- Implications

- Increasing scale and complexity
  - Software black boxes are no longer sufficient
  - **Interactivity** comes more critical to achieve user friendliness, accessibility, and system usability
- Increasing need for software reliability
  - **Diagnosis** and **recovery** for applications failures in addition to debugging becomes critical

Thesis: building an **interactive** and **reliable** operating system platform, **LiteOS**, to simplify operations and improve robustness

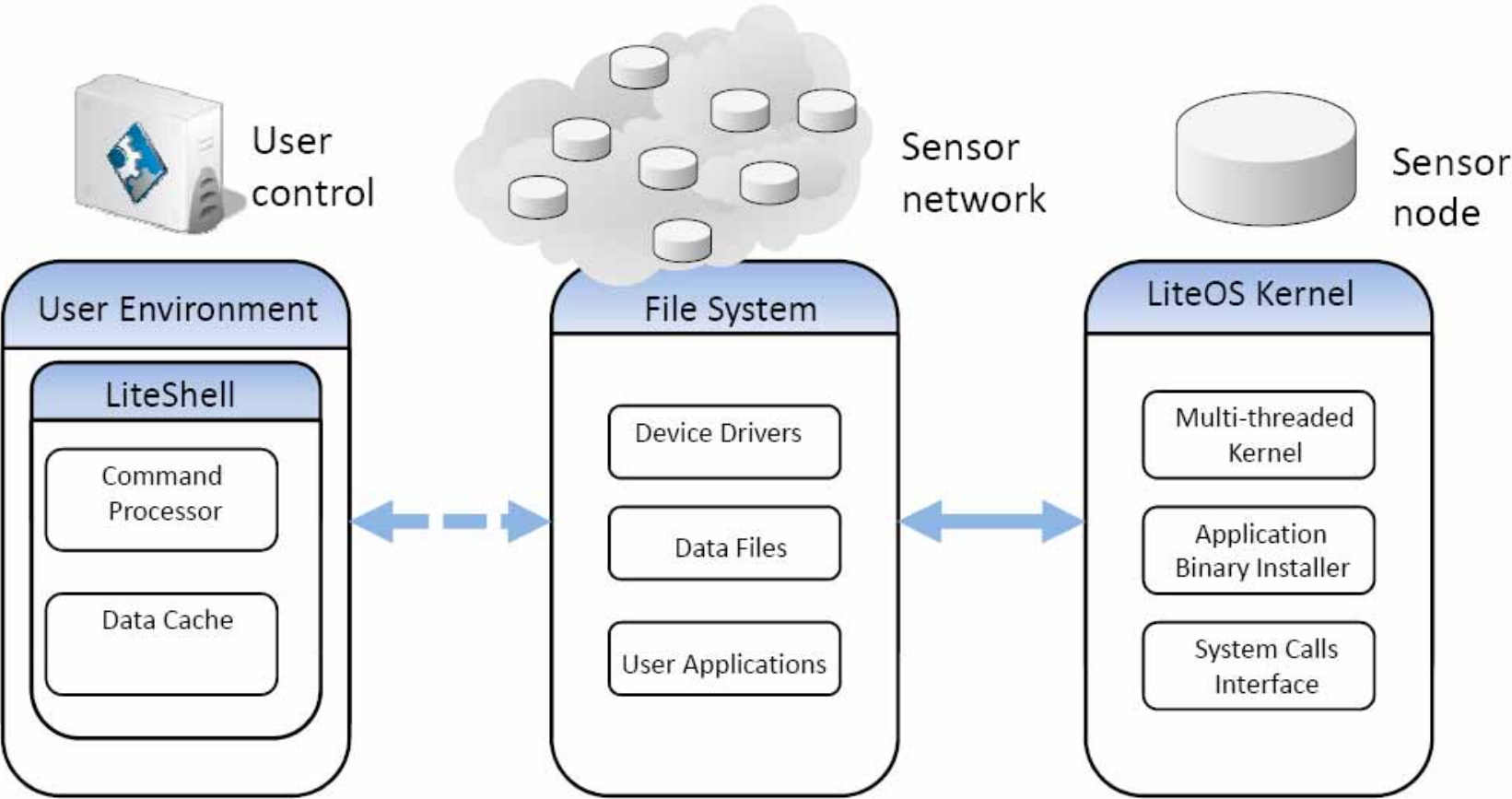
# LiteOS Interactive Model



# Architectural Challenges

- Built with very constrained system resource
  - MicaZ: 4K RAM and 128K Program Flash
  - Low bandwidth on radio
  - Very limited battery power: Two AA batteries
- Scalability and energy efficiency requirement
  - Serious problems because of the scale and lifetime requirement of realistic sensor networks

# LiteOS Implementation Architecture



# LiteOS Reliability

- How to detect and recover from application failures?
- Using memory rules to detect application semantics bugs
  - Rules are extracted from the source code
  - Checked against the runtime state by the kernel
  - **Checkpoint** and **restore** to recover from un-deterministic bugs
  - **Logging** and **offline analysis** for deterministic bugs
- Using file system assisted communication stacks to isolate failures
  - We observe that communication related bugs contribute to many system failures in wireless sensor networks
  - Make different communication stacks exchangeable so that one fault in one stack will not propagate to applications

# Conclusion

- Propose to build an interactive and reliable software platform
  - Improved software accessibility and visibility through Unix-like interactive support
  - Improved reliability through run-time memory checking and dynamically loadable communication stacks
  - Questions?