Wiselib School 2010

Session 2

Hands On

Tobias Baumgartner

Institute of Operating Systems and Computer Networks

October 2010

Task

- Implement a *Remote UART*
  - Model of the *Serial Communication* Concept
    $\Rightarrow$ So that it can be used as ordinary UART
  - Automatically sends UART messages to a sink (where they are passed to the *real* UART)
    $\Rightarrow$ *Real* UART must be passed as template parameter
  - Can accept messages from a sink, and pass to registered callback(s)
- Automatically look for a sink which is connected to gateway
- Use routing algorithm(s) from the Wiselib to realize the task
- Implement as generic Wiselib application, so you can test in Shawn and on iSense

Hints

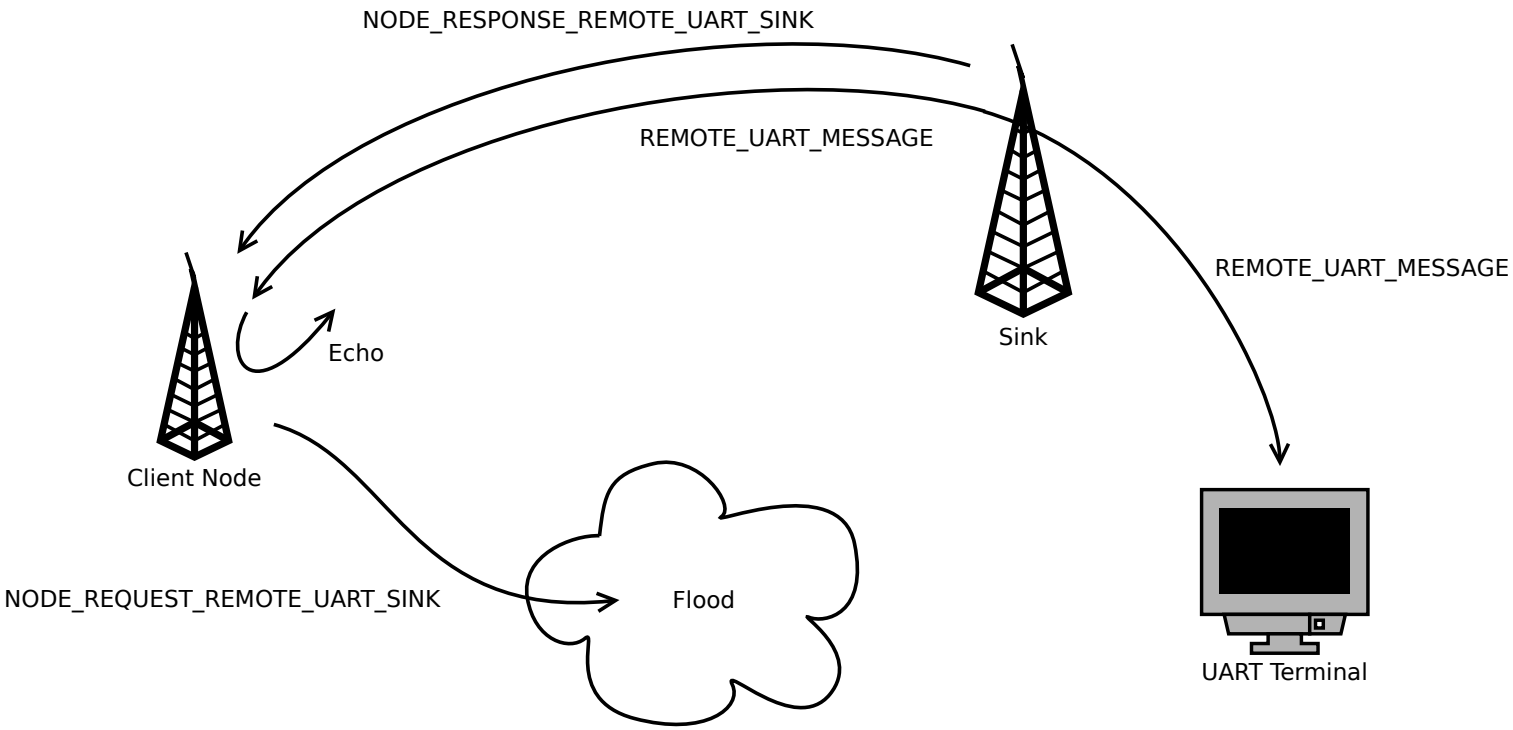
- There is a template provided in `hands-on/session2` that can be compiled for (at least) Shawn and iSense
- Use the provided `Message` class to read/write message headers (then you can test on iSense with our provided gateway!)
- Provided template parameters are only *hints* - feel free to change them for your needs!
- How it **can** be done:
  - Use flooding algorithm to request a sink (search whole network)
  - Use routing algorithm to reply to such a query
  - Use routing algorithm to send remote UART messages when sink is known
- Pass as template parameters, so that algorithms can be exchanged
- For Shawn, there is automatically an UART taken that writes every output to a StringTag that in turn is drawn in Visualization

Remote UART Communication

- UART messages have command type `REMOTE_UART_MESSAGE`
- If the sink receives such a message it should write them **unmodified** (with header) to the UART
- Any other node should just echo such a message back to the sender

- When data from the physical UART comes in, assume it is a `Message`
- If the command type is `REMOTE_UART_MESSAGE`, forward the message to its `destination()` via radio
- In that case, the node can also assume it is the sink

Communication Flow Overview



Helpful Links & Tips

- `Message` is documented in `message.h`
- You only need to edit `remote_uart.h`

Concepts

- Serial Communication
- Radio
- Timer
- Debug