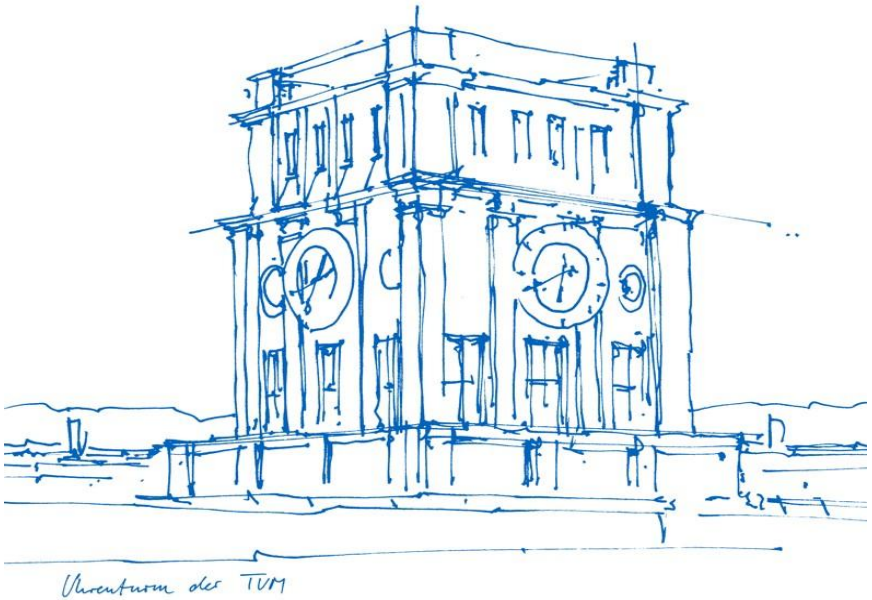
Chair of Communication Networks
TUM School of Computation, Information and Technology
Technical University of Munich

Bedside Patient Monitoring Network

Group 1
Derico Pratama
Abdulrazak Alghabra
Karan Mehta



©2022 Technical University of Munich

Outline

- Introduction
- Environment Conditions
- Network Topology
- Network Architecture
- MAC Configurations
- Patient Mote
- Routing Mote
- Routing Techniques
- GUI
- Results and Discussion

Introduction

- Frequently, the hospitals encounters the issue of **understaffing**.
- With a nurse-to-patient ratio of 1:4[1], patient monitoring systems are crucially important.
- This systems reduces adverse events and mortality rates[2].
- The health condition of the patients will be **recorded by multiple sensors**.
- This sensor data will be then passed through the network to central computer.
- The Central Computer will **monitor the data**, and **raise an alarm** if there are any abnormalities.
- This is will inform a dedicated nurse to make a visit to the following patient.

1. American Association of Colleges of Nursing
2. Journal of Medical Internet Research, Volume 19, Issue 4)

Environment Conditions

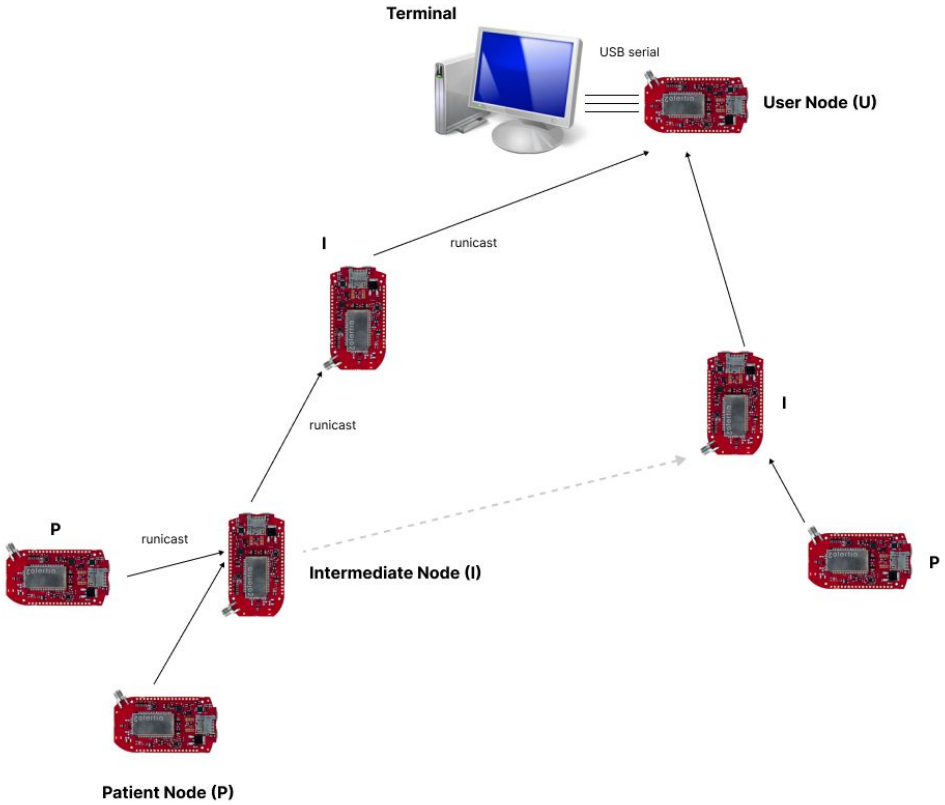
Environment:

- **Stationary**: sensors assumed to stay on the bedside
- **Static**: no frequent topology changes
- **Low latency** is the highest priority

Setup:

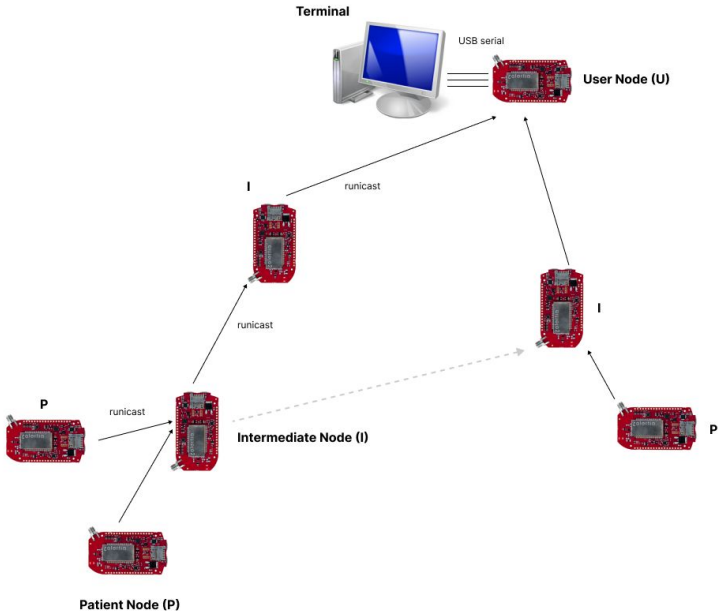
- For representation of our idea, we have considered 3 patients.
- Each patient will have **Heart Rate, and Motion sensor**.
- All these sensors connected to a *Patient Node*.
- These data will be collected and then passed to the Main Computer.
- Main computer will then **monitor the data** and **raise an alarm**.

Network Topology Example



Network Architecture

- Heterogeneous and Hierarchical Network: **Patient (P) nodes → Routing (R) nodes → User (U)**
- Multi-hop networking
- Patient = 0 - 7 dbm, Routers = 7 dbm



Messages

- Two class: 1) **Payload** and 2) **Discovery**

Payload Message =

History= [
1,
2,
3
],

Depth = 3

Bpm = 60,
In_bed = true

Routing Tables =

[
1,2
],
2,3,1
],
3,5,2
],
5,3
]

MAC Configurations

- **Contention-based MAC (CSMA)** → low latency for small to medium load
- **NullRDC** → reduced latency; no power concern since it is stationary and indoor
- **Broadcast + Runicast Tx** →
  - Broadcast: Network Discovery Phase
  - Runicast: Data Transmission Phase; ACK to ensure routes are working

Patient Motes

- First, find a routing mote: send broadcasts and listen for broadcast replies from routing motes (0xFFFF > routing motes id >= 0x8000)
- Increase power until routing mote found/max power reached.
- Periodically check HR readings, interrupt based readings for motion sensor
- Send sensor readings to routing mote via runicast
- In case of failure to send, restart discovery

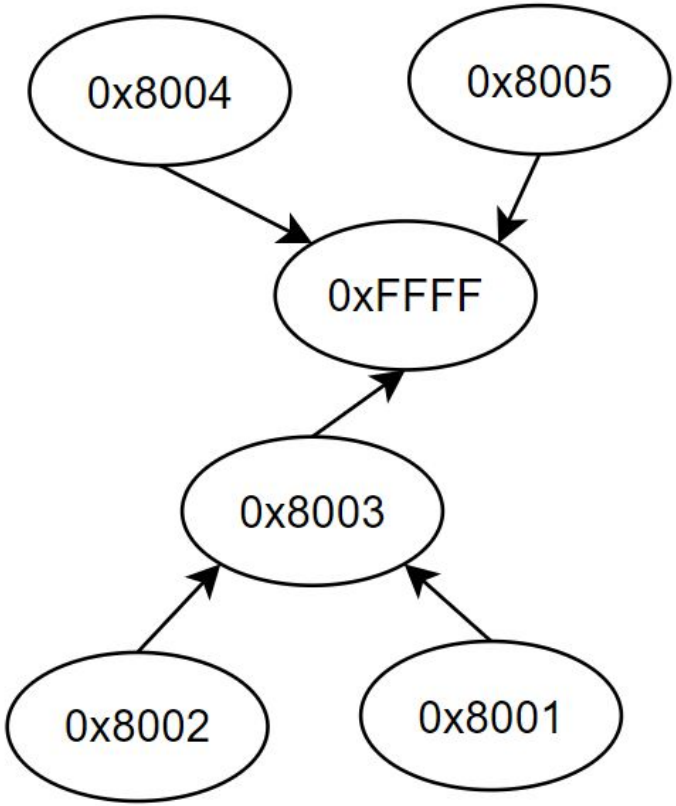
Routing Motes

- First, find neighbors using any broadcast message, add all routing (or user) motes within range to neighbor list
- Broadcast neighbor list, add received neighbor lists to table (database) containing full network, broadcast the table (by flooding)
- Respond to patient broadcast with own broadcast
- Receive runicast messages from routing/patient motes, send messages to next routing mote based on routing algorithm
- Network Discovery refreshes every 120 seconds
- In case of failure to send, rediscover network and broadcast “Reset”. Receiving “Reset” → rediscover network.
- “Reset” also sent upon start to update upon adding new motes
- User: 0xFF, works like routers but only sends neighbor tables and database, only receive routing messages. Does not communicate with the patients. Connected to PC

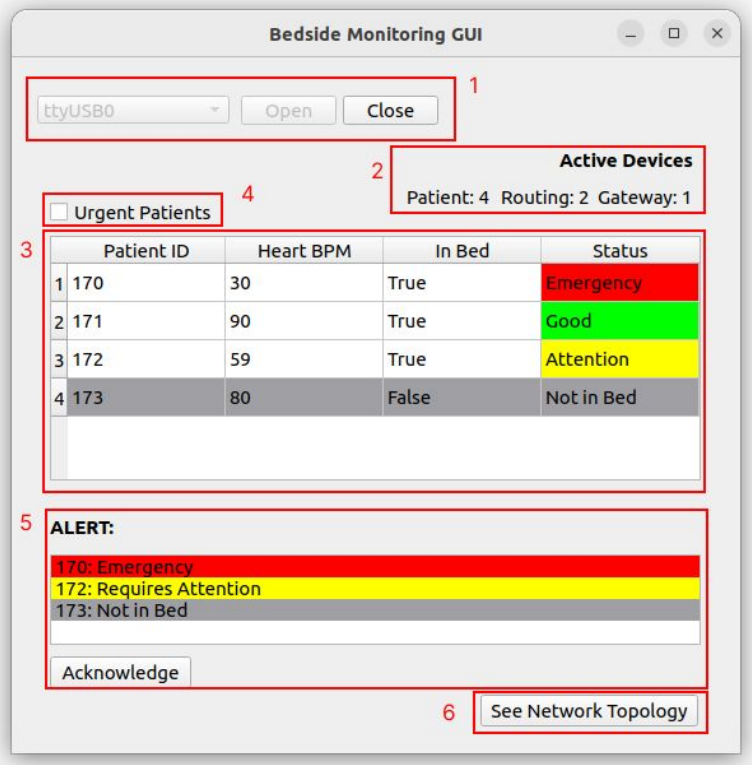
Routing Techniques

- Find Destination in its entry
- If not, find the entry of destination in table
- Check the entries of FFFF
- Check if its neighbour is in these entries
- If found, update its next hop
- Update the number of Hop
- Choose the next hop with minimum Hop

0x8001	0x8002	0x8003	
0x8005	0xFFFF		
0xFFFF	0x8003	0x8004	0x8005
0x8003	0x8001	0xFFFF	0x8002
0x8004	0xFFFF		
0x8002	0x8001	0x8003	



GUI

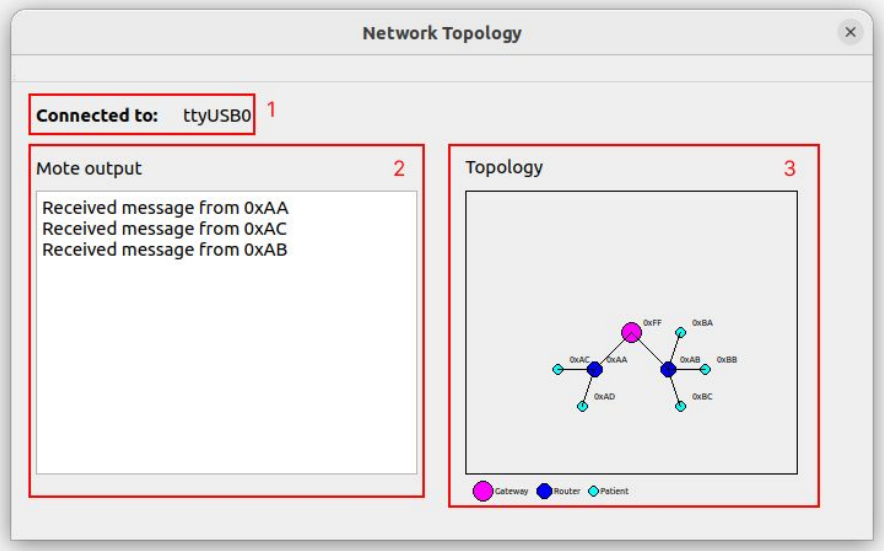


Main Window:

1. Serial port connector
2. Status of active devices
3. Data table
4. Data filter for urgent patients
5. Alert for urgent patients
6. Access to the “Network Topology” page

Time to Live (TTL) → to ensure freshness of data and avoid the table from piling up.

GUI



Network Topology Window:

1. Connection status
2. Mote output terminal
3. Network topology graph

Q&A

Appendix: Routing Mote vs Patient Mote

Feature	Routing Mote	Patient Mote
Network Discovery	Find Neighbors, then share list of neighbors to build a table of all nodes and their neighbors	Search for a Routing Mote, increase power until one is found. If none are found, restart process after time
Failure Recovery	If connection breaks, restart discovery process	If connection breaks, restart discovery process
Function	Receive patient/routing messages, forward them in the direction of the terminal node	Read sensor readings, then send the data in a patient message to a routing mote