KTH Royal Institute of Technology

VHF/UHF Wireless Uplink Solutions for Remote Wireless Sensor Networks

Alp Sayin
22 May 2013
sayin[at]kth[dot]se

outline

- Background
- Problem
- Goals
- Theory
- Design
- Implementation
- Experiments
- Results
- Conclusions
- Future Work
- Q & A

background - general

- Atmel ATMega128RF-chip with IEEE 802.15.4Transceiver as Mote
- The mote software is based on the Contiki operating system.
- A mote automatically becomes a sink mote when connected via a TTL/USB converter
- Gateway is usually a Bifrost/Alix system or Raspberry Pi without internet connection

background - sensors

- Sensors
  - Pressure
  - Temperature
  - Precipitation
  - Soil moisture
  - Solar efficiency
  - Drinking water quality
    - Turbidity
    - Acidity
    - Redox-potential

background - motes

- MCU
  - Atmel ATMega128Rfa1
    - Internal ADC
    - Built-in IEEE 802.15.4 Radio Transceiver
- OS
  - TinyOS
    - NesC
  - Contiki-Os
    - C99

background - communication

- Inter-node communication
  - IEEE 802.15.4
    - 10-20 meters range
    - 250 kbit/s raw data rate
  - Contiki Rime Protocol
- Sink-Gateway communication
  - RS232 Uart Communication
  - Sensd by Robert Olsson and Jens Laas

background - gateway

- Sink Node
  - Any Herjulf mote can be used as a sink node
    - No extra programming required
- Gateway
  - Alix 2d13 machine
    - Bifrost Os
    - Voyage Linux

background - uplink

- Uplink
  - Cabled connection
    - Fiber optic
    - Ethernet cables
  - Wireless connection
    - GSM network (GPRS, HSDPA etc.)
    - Satellite link
    - Dedicated terrestrial wireless link on a suitable frequency
  - Physical transportation of data
    - Delay tolerant network

problem

- Get the collected data out from the gateway of a WSN to a remote repository with internet access.
- Explore 434 MHz and 144 MHz frequencies and associated protocol stacks to optimize the range and QoS
- Try from dedicated hardware solutions to software defined radio links to optimize power consumption and flexibility.

goals

- A feasible, minimum hardware, low-cost, low-power, long range solution to the problem of getting the sensor data out from the sink mote to a remote repository
- To implement IP over the same solution, so that existing user programs, or newly developed programs could be used over this link

theory – physical link

- VHF / UHF Bands
  - 144 Mhz Band (2 m band)
  - 433 Mhz Band (70 cm band)
- Modulation
  - Bi-phase encoding (Manchester)
  - 1200 baud AFSK

theory – data link

- ONLY FRAME FORMAT!
  - Ethernet
    - Max payload 1500 byte
    - 18 bytes header
      - 6 byte source and destination
      - 2 byte length
      - 4 byte CRC checksum
  - IEEE 802.15.4
    - Max payload 256 byte
    - 21 bytes header
      - 2 byte frame control
      - 1 byte sequence number
      - 4-20 byte address information
      - 2 byte frame check sequence
  - AX.25
    - Max Payload 256 byte
    - 18 bytes header
      - 7 byte source and destination
      - 1 byte control
      - 1 byte protocol identifier
      - 2 byte CRC checksum

theory – network layer

- IP
  - IPv4
    - 20 byte header
      - 4 byte source and destination
      - 2 byte header checksum
      - 2 byte total length
      - 1 byte time-to-live
  - IPv6
    - 40 byte header
      - 16 byte source and destination
      - 2 byte payload length
      - 1 byte time-to-live
- APRS
  - Rests on AX.25
  - Text based protocol
  - No header requirement except AX.25 headers

theory – transport layer

- UDP
  - Easy to implement, difficult to use
- TCP
  - Difficult to implement, easier to use
- APRS
  - Pseudo-routing routines are implemented
  - Based on broadcast flooding with reverse poisoning

theory – application layer

- HTTP
  - Simple request/response protocol
  - Can be used for any resource access application
  - Requires TCP
- TFTP
  - An old yet reliable protocol (still used in some routers for flashing)
  - Can work on UDP
  - Has limitations (e.g. file size < 4 GB)
- APRS
  - Generally used for weather and location reporting
  - Also has a telemetry message option for reporting arbitrary measurements
  - Messages are collected in a single open repository (accessible via aprs.fi)

theory - hardware

- TNC with Radio rigs
  - Radio
    - MAAS AHT-UV-2
    - YAESU FT8900
  - TNC
    - Hardware TNC
    - Software TNC
      - soundmodem
- Dedicated Radio modules
  - Radiometrix Bim2A
  - Radiometric UHX1

design - protocols

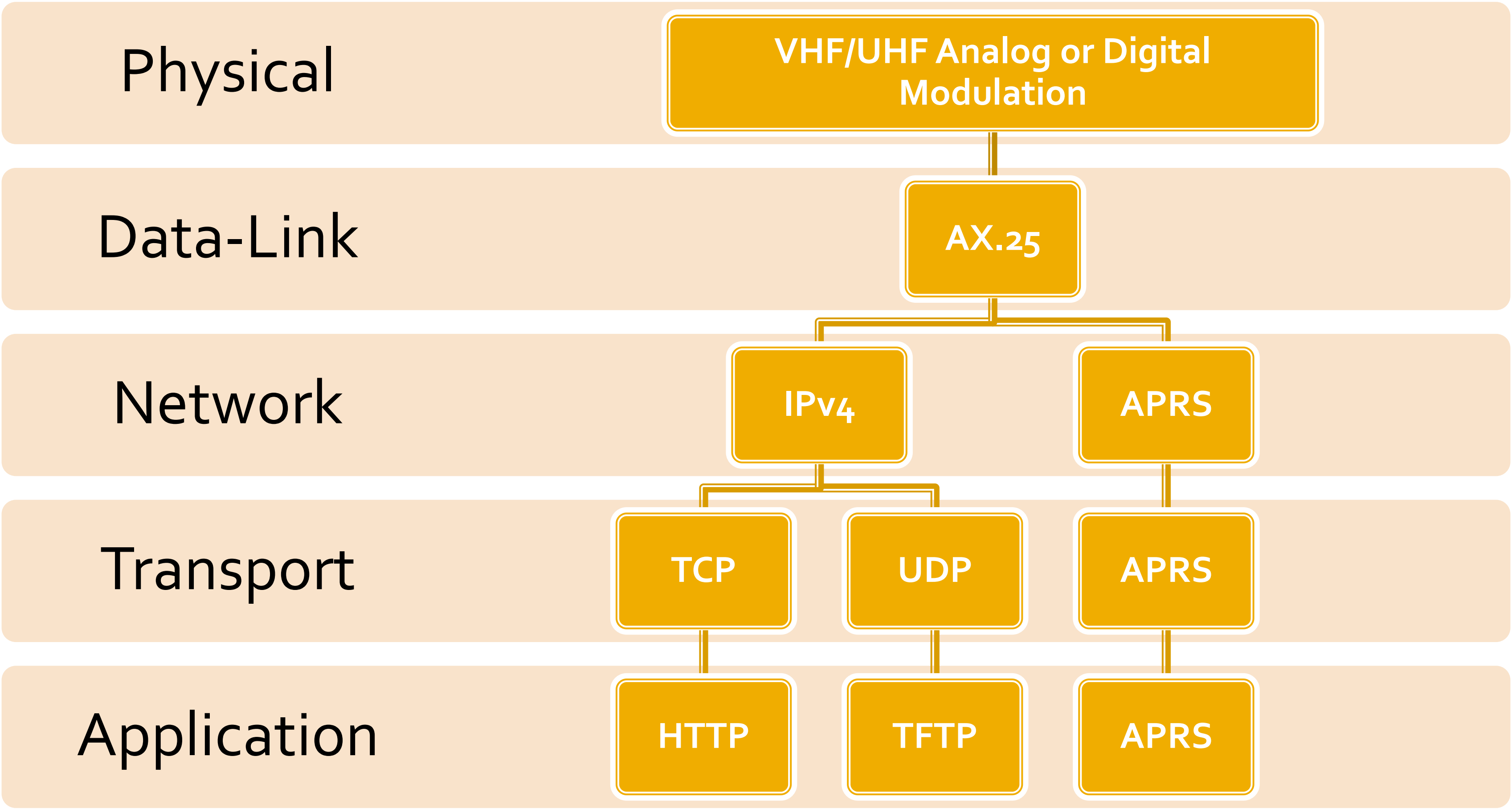


Figure.1 Resulting branches in the project after design decisions

design - final

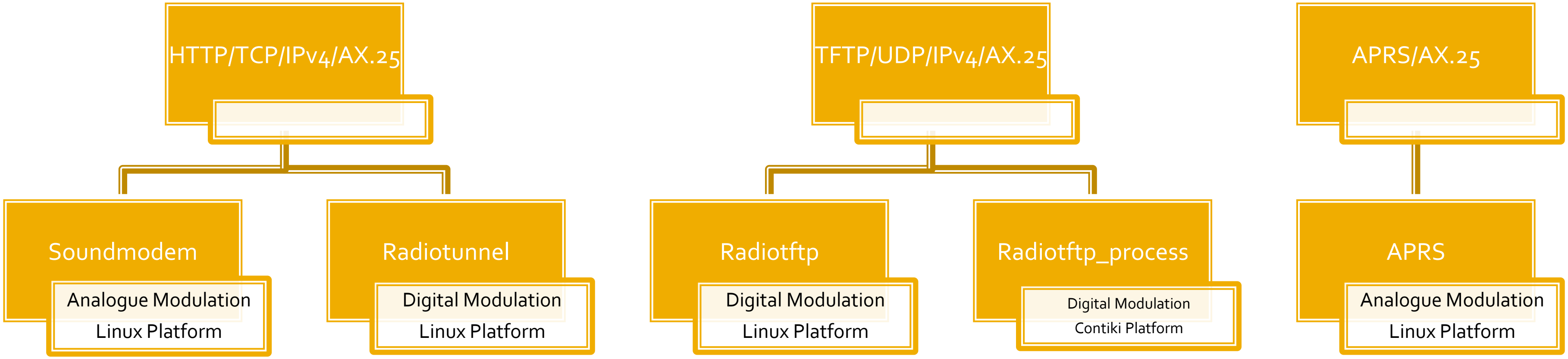


Figure.2 Resulting solutions after hardware decisions

implementation - radiotftp

- TFTP < UDP < IPv4 < AX.25 < Bi-phase Encoded Digital Stream
- Tailored solution for the problem
- Completely in C, easier to port

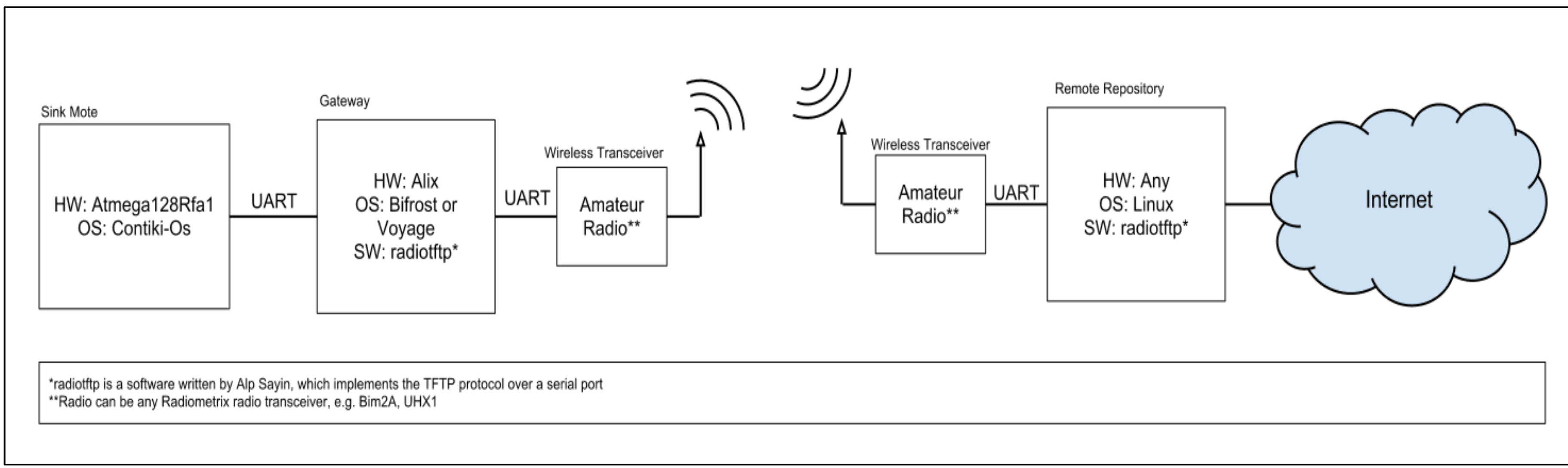


Figure.3 System diagram for radiotftp solution

implementation – radiotftp_process

- TFTP < UDP < IPv4 < AX.25 < Bi-phase Encoded Digital Stream
- Tailored solution for the problem
- Contiki-Os over Atmega128Rfa1 port of radiotftp

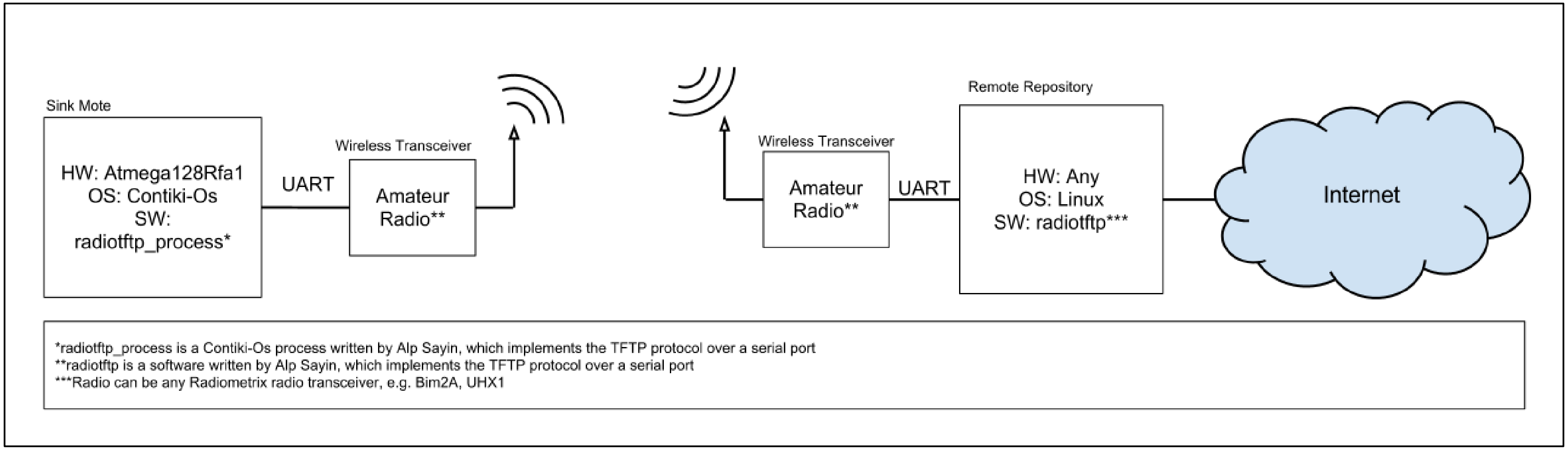


Figure.4 System diagram for radiotftp_process solution

implementation - radiotunnel

- IPv4 > AX.25 > Bi-phase Encoded Digital Stream
- Flexible solution, open for any application layer compatible with UDP or TCP
- Easy to implement with use of virtual kernel interfaces

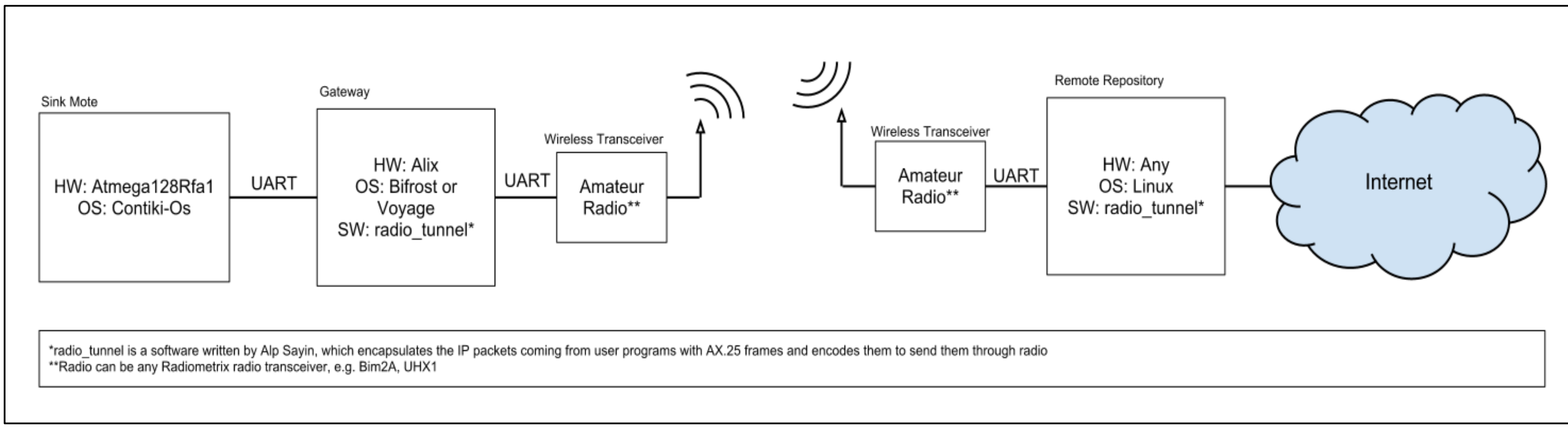


Figure.5 System diagram for radio_tunnel solution

implementation - soundmodem

- IPv4 > AX.25 > AFSK Analog Stream
- Flexible solution, open for any application layer compatible with UDP or TCP
- Software is already there
- Requires more hardware

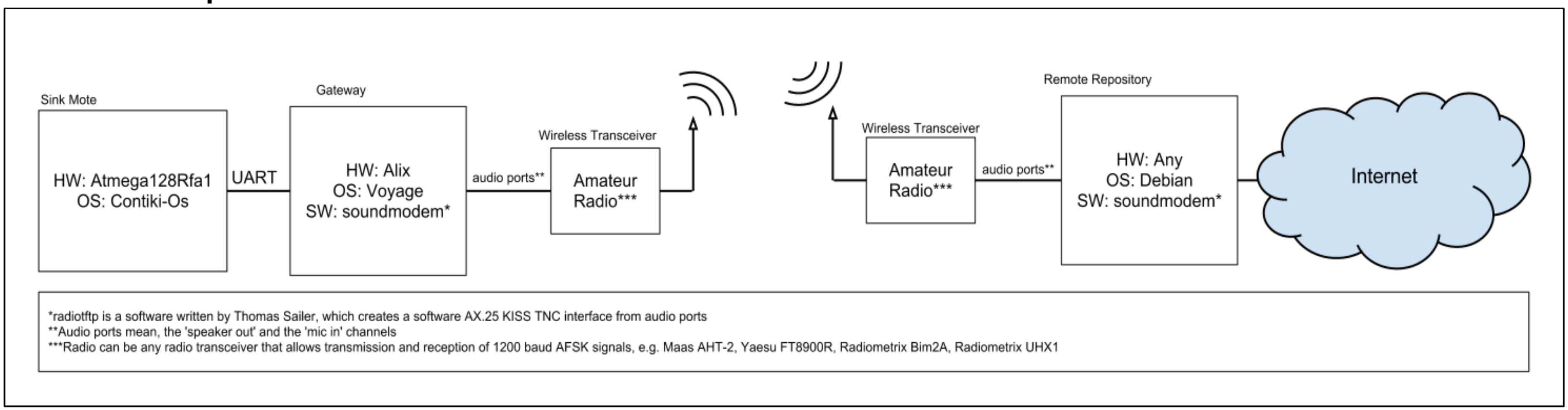


Figure.6 System diagram for soundmodem solution

implementation - aprs

- APRS > AX.25
- Utilizes the APRS network
- Limited by APRS and radio amateur rules

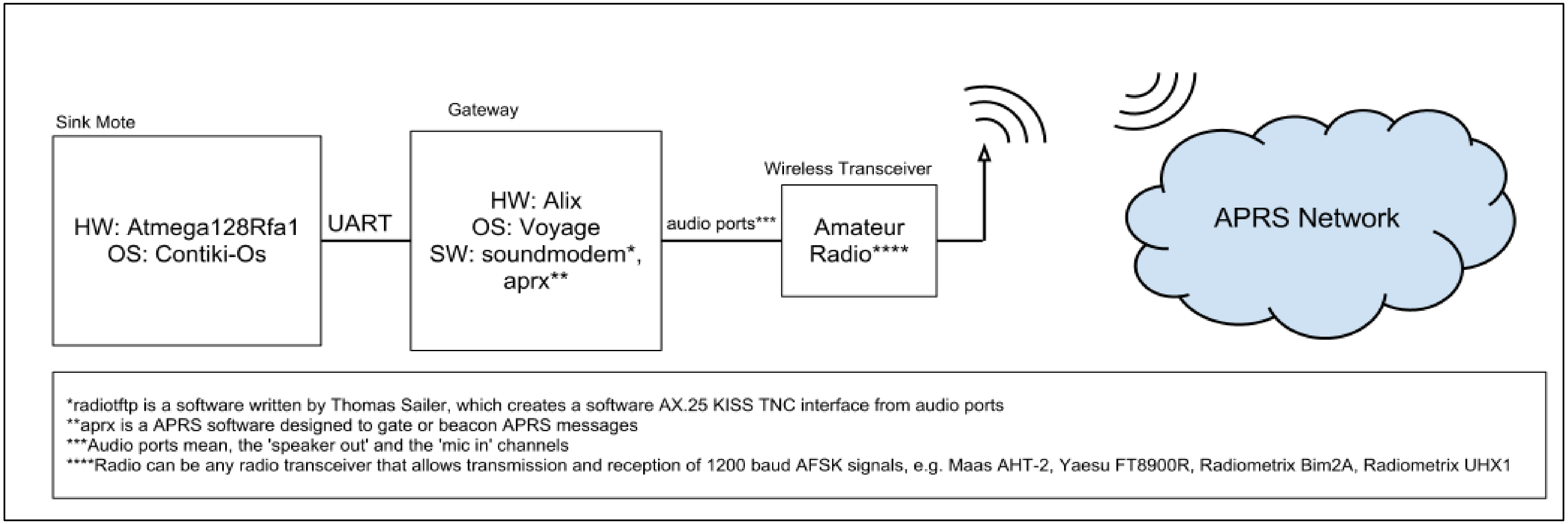


Figure.7 System diagram for APRS solution

experiments

- Radiotftp
  - Parameters: Distance, Transfer Size
  - Outputs: Transfer Time, Packet Error Rate, Power Consumption
- Radiotunnel
  - Parameters: Transfer Size
  - Outputs: Transfer Time, Channel Utilization, Power Consumption
- Soundmodem
  - Parameters: Transfer Size
  - Outputs: Transfer Time, Channel Utilization, Power Consumption
- General Experiments with Radiometrix Devices
  - Parameters: Distance
  - Outputs: Received Signal Strength Indicator (RSSI)

experiments – outdoor map

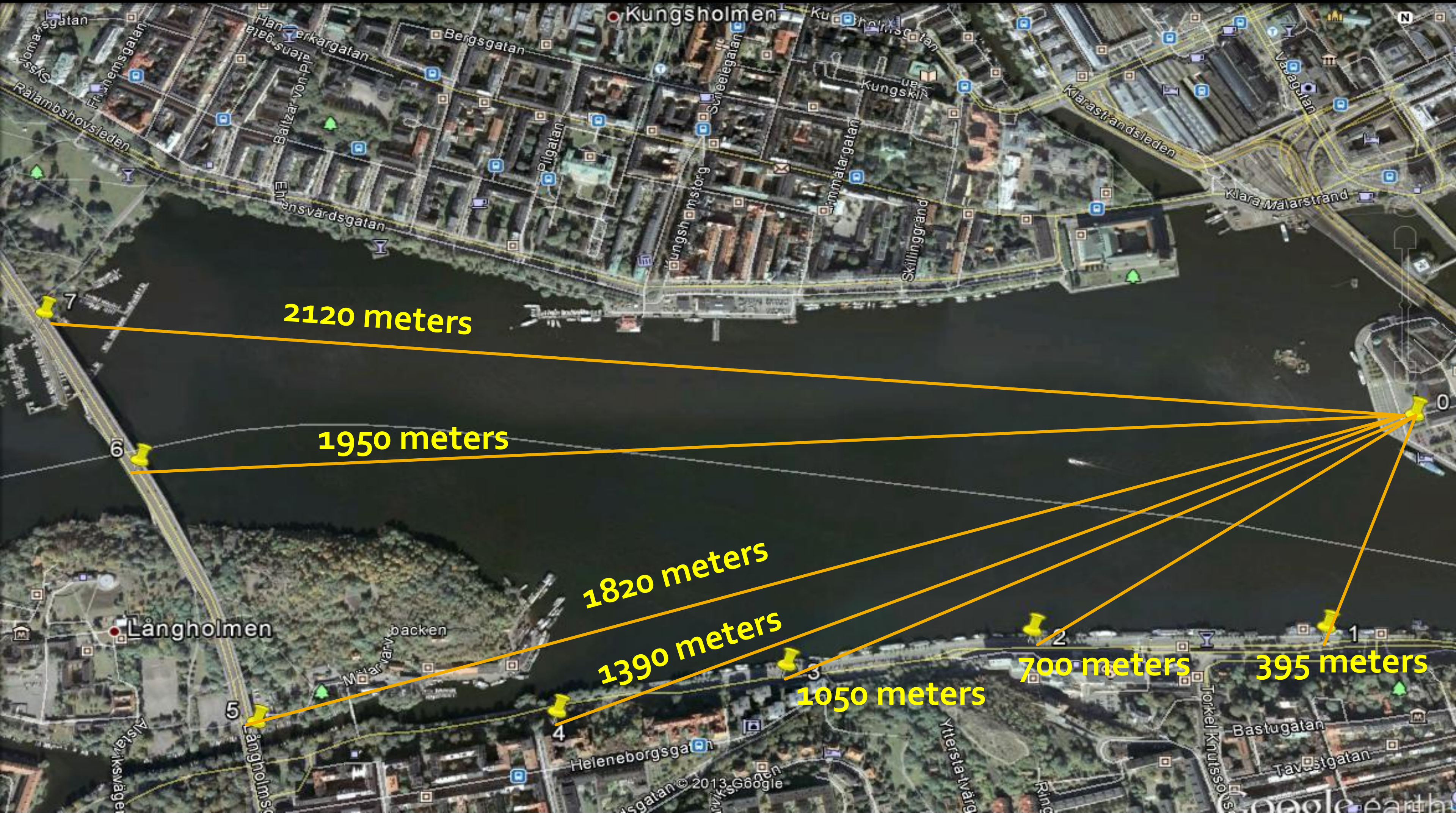


Figure.8 Map of Gamla Stan Experiments (image from Google Maps)

results

	Transfer Time 127 bytes	Transfer Time 2 kbytes
radiotftp uhx1	00:08.915	00:21.727
radiotftp bim2a	00:00.873	00:02.414
radiotunnel uhx1	02:56.029	12:09.429
radiotunnel bim2a	02:00.120	02:05.261
soundmodem	02:09.707	02:59.324

Table.1 Average transfer times with minimum distance between transceivers

results – transfer times

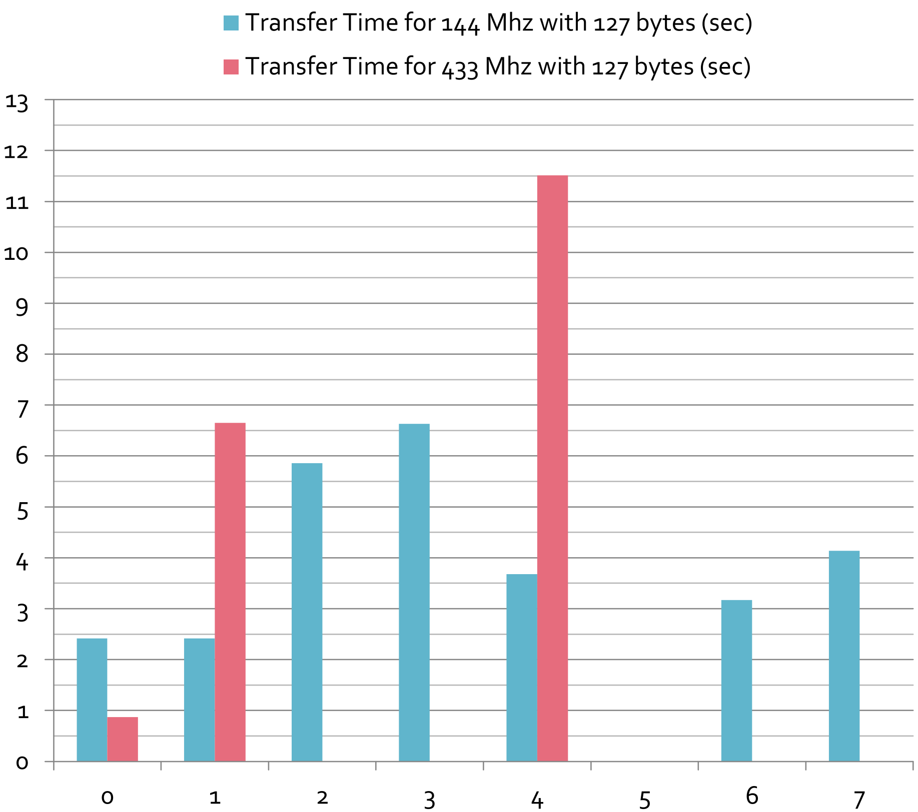


Figure.9 Transfer time plots for single packet delivery

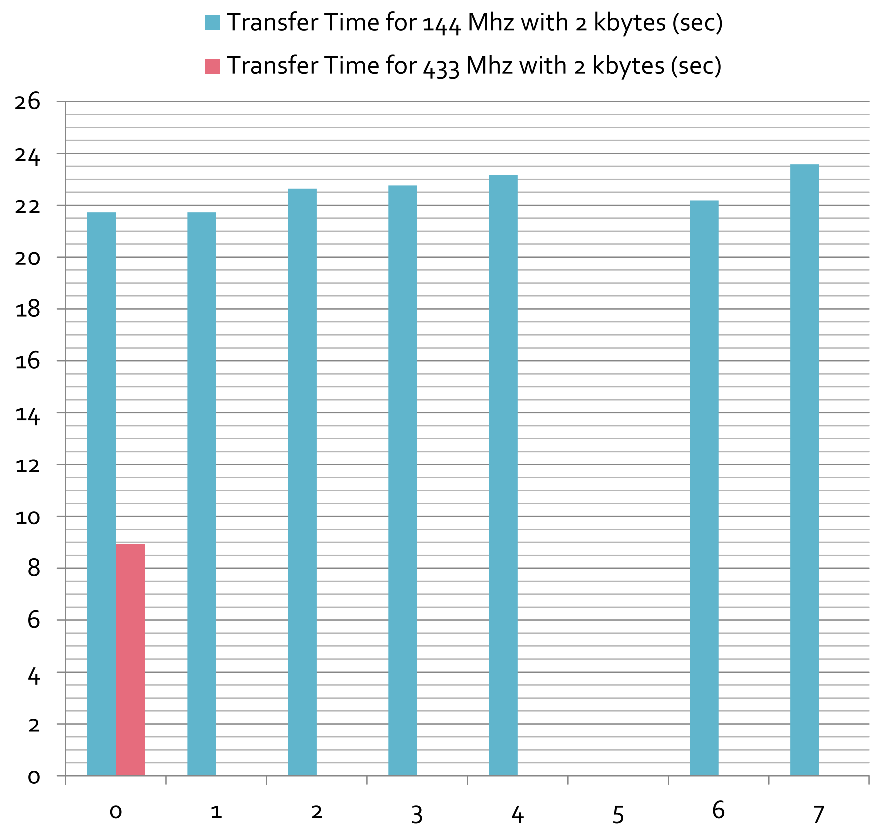


Figure.10 Transfer time plots for many-packet delivery

results – packet error rate

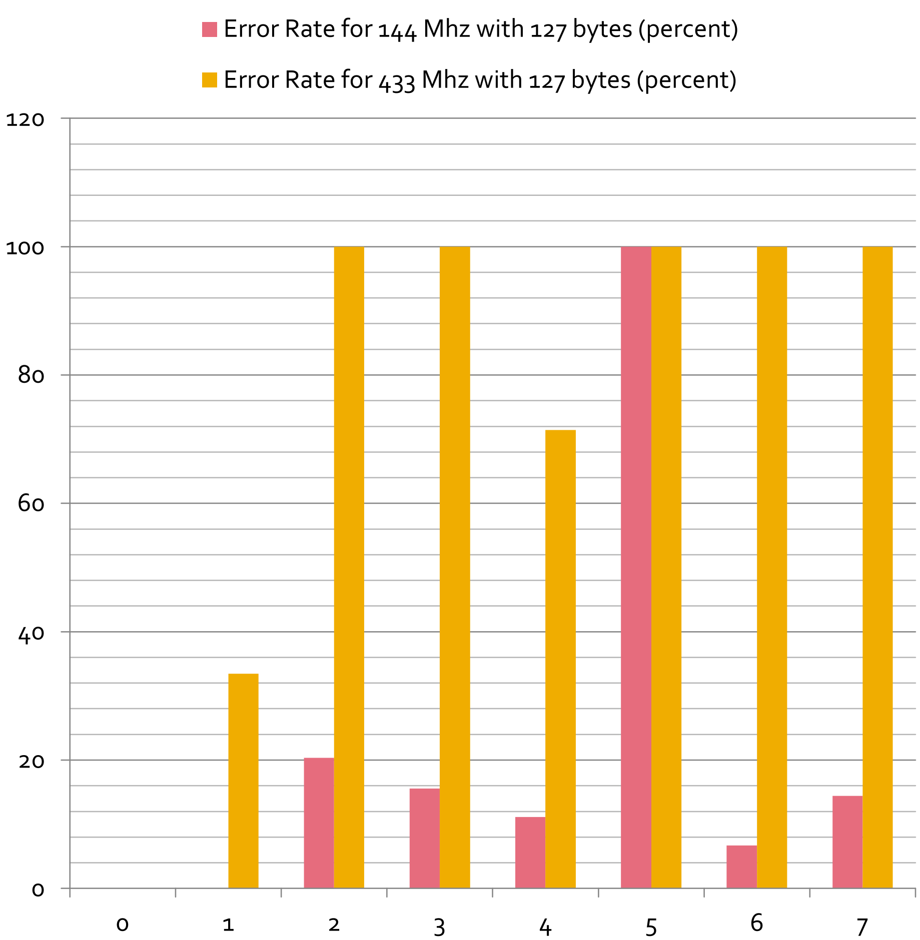


Figure.11 Error rate plots for single packet delivery

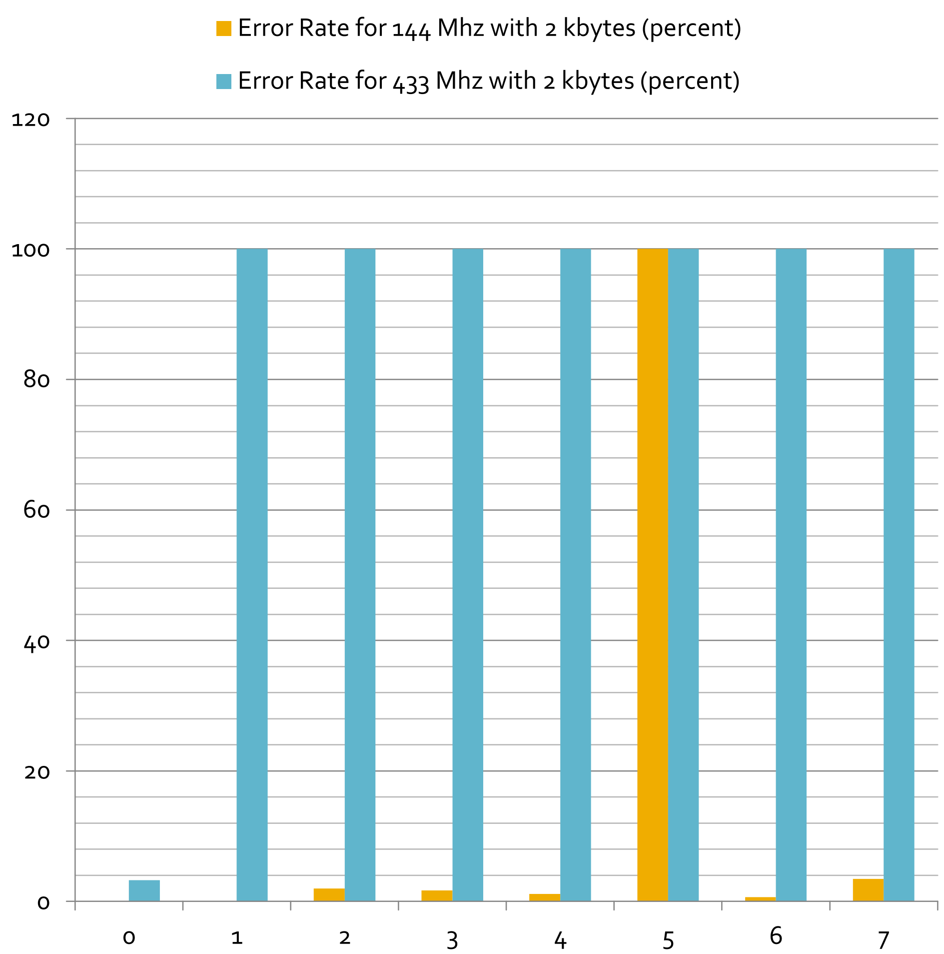


Figure.12 Error rate plots for many-packet delivery

results - bitrate

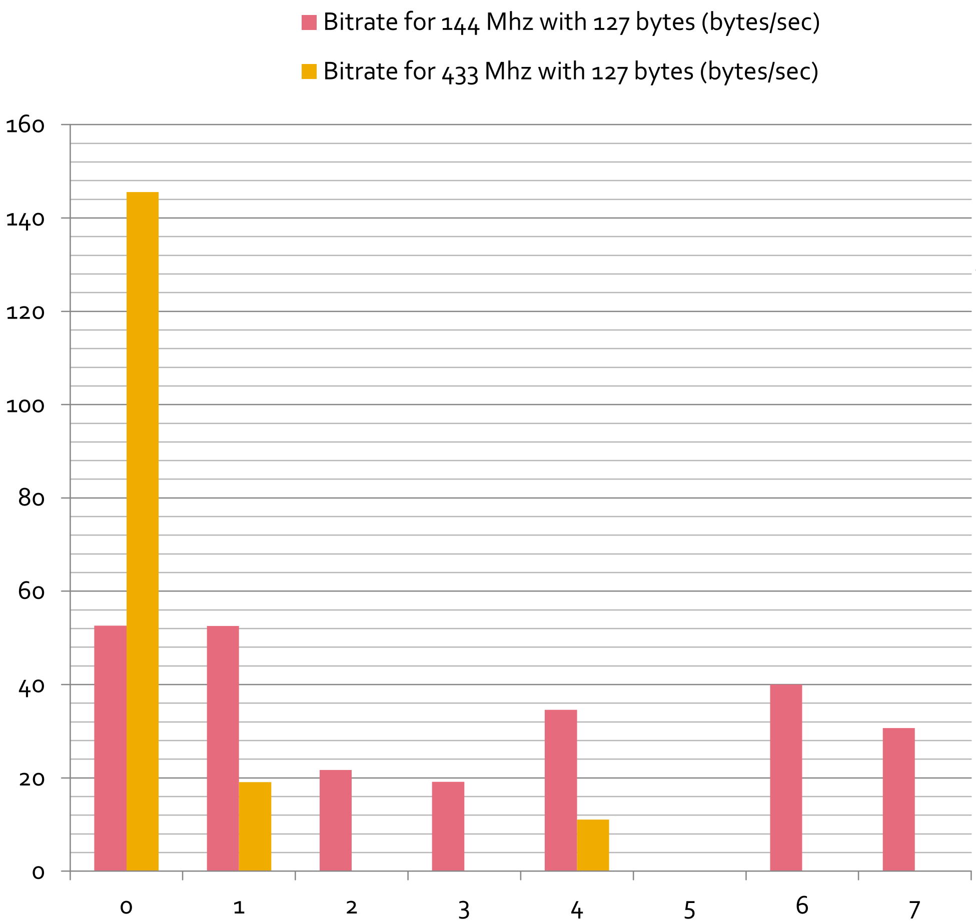


Figure.13 Bitrate plots for single packet delivery

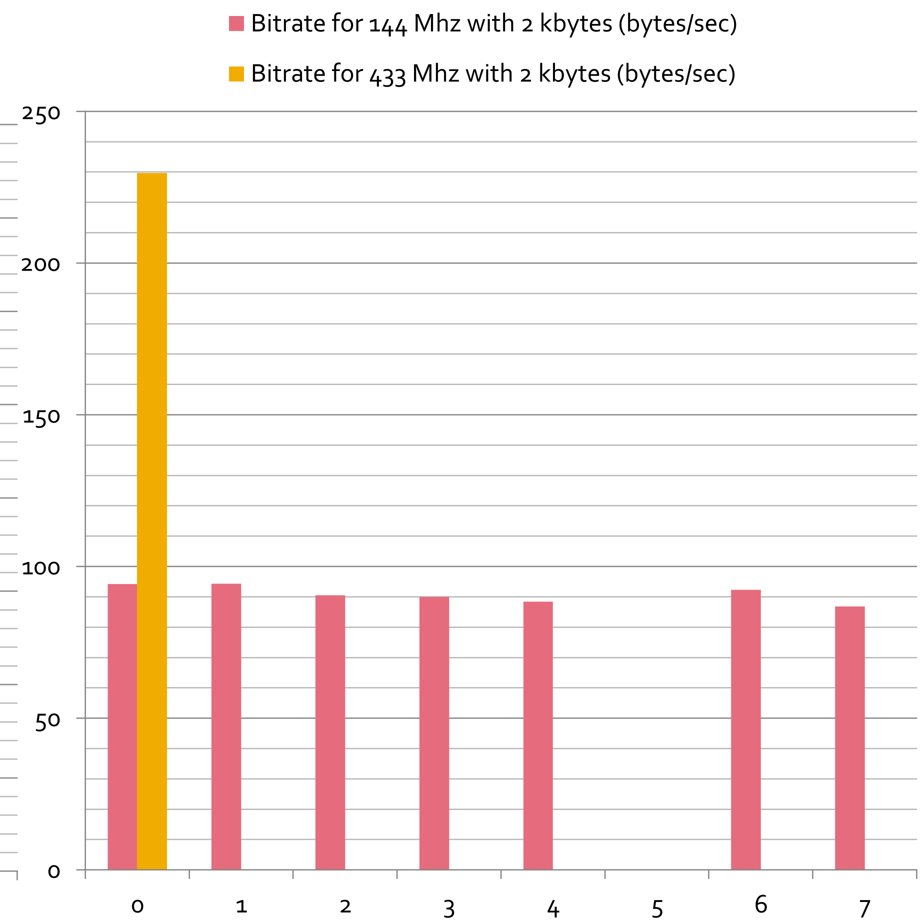


Figure.14 Bitrate plots for many-packet delivery

results – RSSI

Location	RSSI (UHX1)	RSSI (Bim2a)
No Signal Applied	64	0.28
0 (Base)	196	0.48
1	136	0.40
2	122	0.30
3	116	0.31
4	133	0.38
5	103	N/A
6	126	N/A
7	121	N/A

Table.2 RSSI readings from various locations with UHX1 and Bim2A

conclusions - radiotftp

- The effect of overhead can be heavily observed. On the other hand, many-packet transactions are statistically more probable to disconnect ('Experiments with radiotftp').
- The bitrate difference in bands shows itself also in the final throughput.
- While using the 2 meter band, the distance does not seem to have much effect. On the other hand, obstructions on the wave path cause a lot of distortion.
- While using the 70 cm band, the received power decays much more relative to the 2 meter band and therefore observed to have a much shorter range.
- In both bands having a high ground has a good impact on signal strength.

Conclusions - general

- Radiotftp solution seems to have much greater bitrate compared to others, but this is simply an effect of utilizing the channel more efficiently. On the other hand, other solutions can't use the channel this efficiently, even if they wanted to.
- The radiotunnel solution shows almost an exponential growth in transfer time with respect to the file size. This is due to the manual forced drop of the packets to ensure half-duplex operation.
- Soundmodem proved itself to be a faster option compared to radiotunnel, even with its low raw bitrate (1200 bps).
- If the radiotunnel is not to be improved to act as an half-duplex interface, and if soundmodem solution can be improved to use radiometrix devices, then radiotunnel solution can deprecated.
- Some suggestions could be made according to some requirements:
  - If higher throughput is required; radiotftp,
  - If easy-setup and easy API is required; radiotunnel
  - If standardization and easy API is required; soundmodem
  - If standardization and set-it-and-forget-it kind of application is required; APRS solution would be suggested.

future work - radiotftp

- The radiotftp code base should be improved to have multiple-size queues and multiple timers. It already supports it, but it is not a default.
- The radiotftp code should be cleaner for further developers. It should look like an open API -which, it is-, and there should be an open documentation for it.
- The radiotftp solution can be optimized more by changing the place of the delays and/or replacing the delays with non-busy waits.

future work – radiotftp_process

- The radiotftp_process code could be optimized to work with higher baudrates than 2400. So that it would consume less CPU time while transmission.

future work - radiotunnel

- The radiotunnel code should not be improved anymore, but instead, an actual device driver should be written for fine tuning.

future work - soundmodem

- The soundmodem solution should be moved on to work with Radiometrix devices. In such way, a more portable hardware can be obtained. And the users won't need to worry about handheld radios' power consumption.

future work - APRS

- If possible, the APRS solution should also be tested in open field and packet loss should be recorded.

future work - general

- The uhx1_programmer can be extended to be able to program the frequency of the UHX1 devices, so that frequency can be selected while running to avoid interference.
- The devtag library could be incorporated to use USB device tags instead of USB device names to select the proper USB FTDI device.
- A team has already started working on an implementation of a Delay Tolerant Network (DTN) based on this project's outcomes.

questions & answers

- Thank you for listening
- More information:
  - http://alpsayin.com/vhf_uhf_uplink_solutions_for_remote_wireless_sensor_networks
  - http://github.com/alpsayin
  - http://code.google.com/p/kth-wsn-longrange-radio-uplink/ (old)
  - sayin[at]kth[dot]se
- WSN Team 2012
  - http://ttaportal.org/menu/projects/wsn/fall-2012/
  - https://github.com/organizations/WSN-2012
  - https://docs.google.com/presentation/pub?id=1rL40Es9D6Z0AD4bN72XcnrYqhL56eWsP8E4WOMR8C-E&start=false&loop=false&delayms=3000

extra - future work

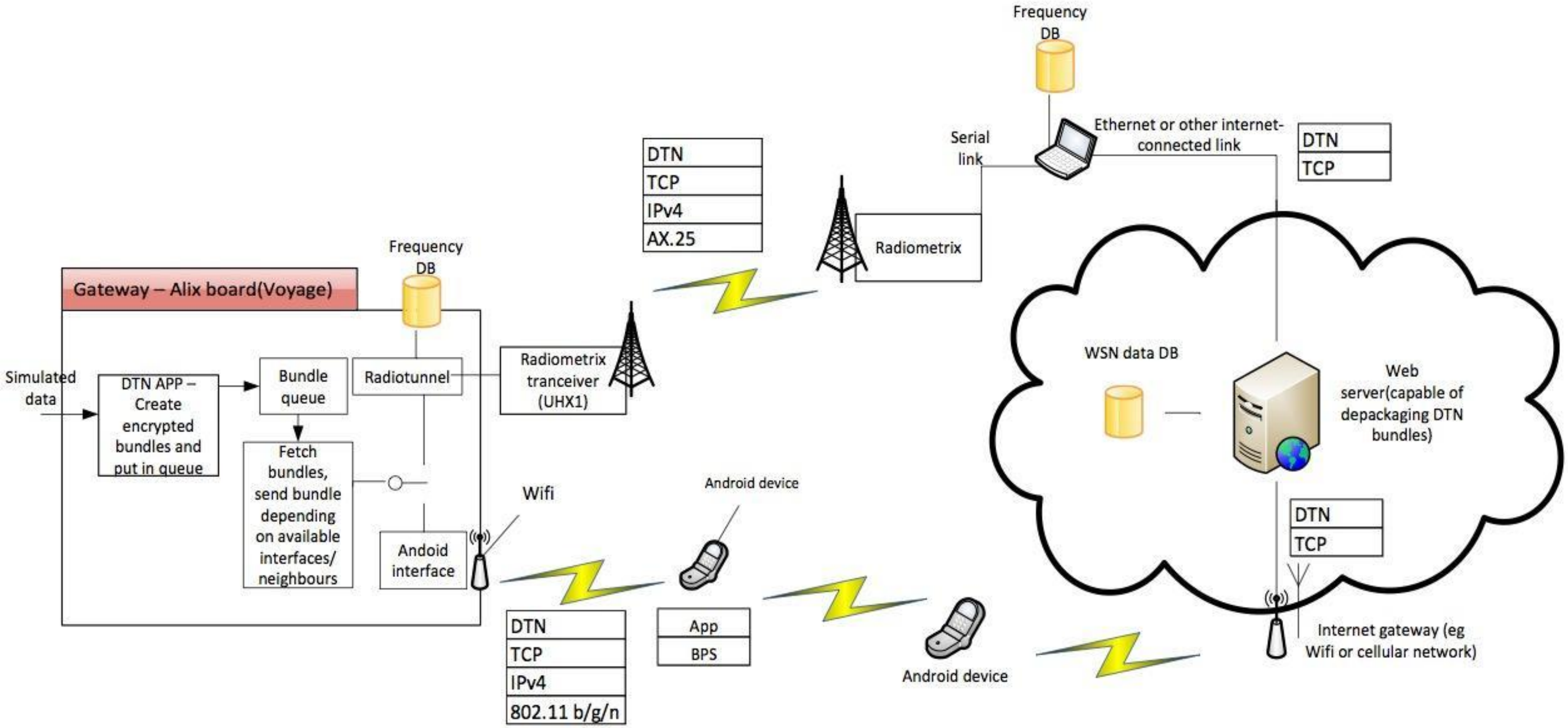
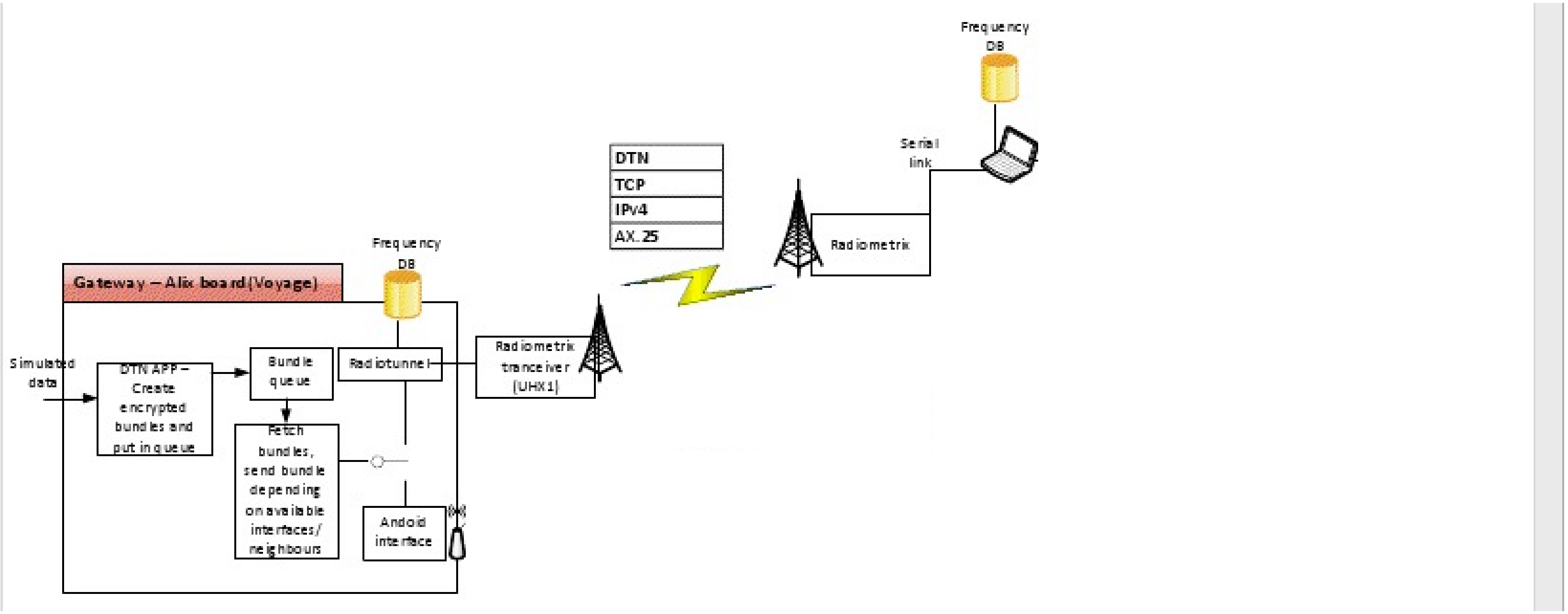


Figure.15 Topology of a VHF Link with a Delay Tolerant Network as fallback
*Courtesy of WSN Team 2012 (KTH Communications System Design, Design Project Team)

extra - delay tolerant network



- Spectrum Database Radio(SDB) Solution
- Selection Mechanism Implementation

Figure.16 Topology of VHF link part of the WSN Team's project
*Courtesy of WSN Team 2012 (KTH Communications System Design, Fall 2012 Design Project Team)

extra - demo

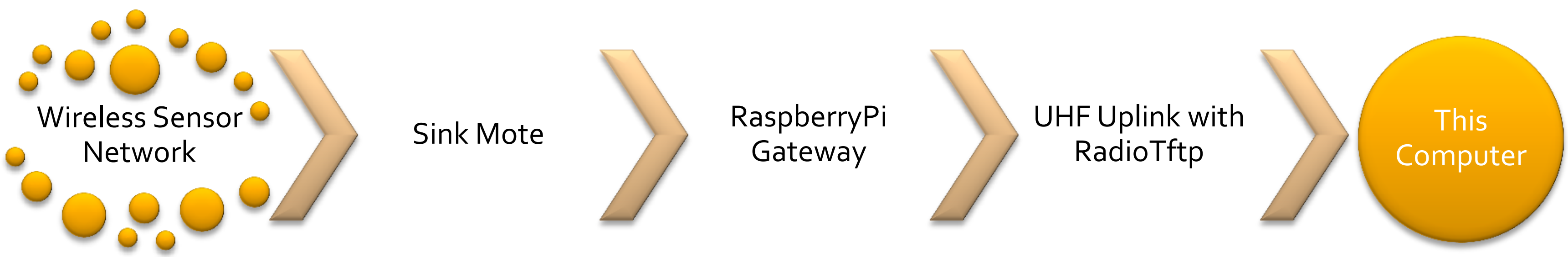


Figure.17 Topology of the demo setup