Building to Building Communication over Wireless Sensor Networks

B. Sree Charan Teja Reddy

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद

Indian Institute of Technology Hyderabad

Department of Electrical Engineering

July 2015

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

B. Charan R.

(Signature)

B. SREE CHARAN TEJA REDDY

(B. Sree Charan Teja Reddy)

EE12M1010

(Roll No.)

Approval Sheet

This Thesis entitled Building to Building Communication over Wireless Sensor Networks by B. Sree Charan Teja Reddy is approved for the degree of Master of Technology from IIT Hyderabad

(Dr. Sumohana Channappayya) Examiner
Dept. of Electrical Eng
IITH

(Dr. P. Rajalakshmi) Examiner
Dept. of Electrical Eng
IITH

(Dr. G. V. V. Sharma) Adviser
Dept. of Electrical Eng
IITH

(Dr. Zafar Ali Khan) Chairman
Dept. of Electrical Eng
IITH

Acknowledgements

I sincerely thank my advisor Dr. G. V. V. Sharma. I benefited a lot from discussions with him. He had always believed in me and supported my ideas. I also thank all my professors from the department and other disciplines who inspired me through their presence.

Dedication

I dedicate this thesis to my parents and to all my friends.

Abstract

The smartphone usage in the low-income sections of India, which form over 70% of the population, is relatively small in comparison to urban areas. Recent reports says that India's mobile penetration is at 73% (up from 30% in 2009), fast catching up to the global average of 93%. India is the fastest growing country for mobile Internet users with the latest predictions at 20% growth every quarter. This growth is mainly driven by low-cost smartphones, which are particularly interesting due to their enhanced capabilities and potential to enable services for the masses. With the advent of affordable smart-phones (Rs 3,000 - Rs 10,000) designed for the Indian user from indigenous manufacturers such as Micromax, Karbonn, etc., more people are shifting to smartphones. Other factors contributing to this shift includes the fact that 98.9% of India currently has no access to fixed-line broadband connectivity options and industries like banking, retail, entertainment, health and education have begun to introduce compelling mobile services. Due to these factors, smartphone sales have begun to cannibalize the traditional feature phone market, while the overall market grew by 12% last year, the smartphone market grew by 229%. At this rate, the smartphone market will overtake feature phones in terms of number of units sold, by 2017.

In the rural areas of India, majority section of the people have low income and poor paying capacity for mobile network operators in regular intervals. They are suffering from inadequate power supply also. A network of low cost and power effective, which can cover an entire village will be very handy to this situation because local people can communicate among themselves with free of charge. Such kind of networks are developed in the previous literature, but using proprietary hardware. So, a network of low cost and power effective supporting voice communication and developed using popular and cheaply available devices like smartphones (Android devices) in the market is desirable.

As a first step towards this we implemented a handset interfacing Android device with IEEE 802.15.4/Zigbee mote. The advantages with this handset are, it is using Android device which is most popular and available at affordable prices and it is using IEEE 802.15.4/Zigbee mote which can be connected to the Zigbee network which is low cost and power effective. Using this handset we implemented peer to peer message chatting application, peer to peer real time half-duplex voice calling application and non-real time voice communication over a static network.

We also analyzed a self-powered IEEE 802.15.4 based multi-hop network consisting of an Internet gateway (IG) and relay nodes for providing Internet facility in rural areas like villages of India. In case of proportional fair scheduling, it has already been shown in the existing literature that if the backhaul capacity of both IG and relay nodes is infinite then it results in users being given equal time in scheduling. However, in case of the proposed multi-hop network, backhaul capacity of both IG and relay nodes is limited. Hence, we analyze the scheduling for the proposed multi-hop network in presence of limited backhaul capacities.

Contents

Chapter 1

Android controlled Zigbee motes for Wireless Sensor Networks

1.1 Introduction

Wireless Sensor Networks (WSNs) are primarily used in applications where low bit rate, low cost, low complexity and more lifetime are required. WSN nodes usually consist of an 8 or 16 bit microcontroller and a short range transceiver in order to create a low cost network with maximum lifetime. Traditionally, WSNs have been used for sensing and reporting. Recently, however, there is a great demand to support a short range communication system in WSNs for use in educational institutes, residential areas and even in emergency situations. Voice over WSNs was implemented in [1] and [2]. However, these implementations were done on proprietary hardware. But, for this project we used the android platform along with Zigbee protocol both of which are quite popular and open in nature.

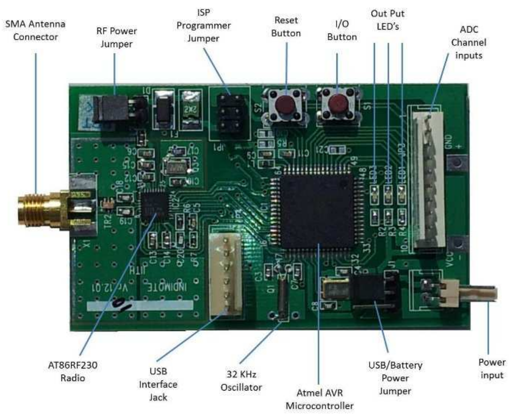

Figure 1.1: IITH-mote

In this project, we use IIT-H motes as nodes in the network (see Fig. 1). IITH-mote is a 2.4 GHz mote used for enabling low-power WSNs. It works on 8 MHz atmega microcontrollers with 8 KB RAM. It has a 6 pin usb interface jack for serial communication and uses IEEE 802.15.4 [3] compliant RF (Radio Frequency) Transceiver for radio communication. IIT-H mote supports two ways of communication. One is wireless communication through RF transceiver. The other way is wired serial communication through Universal Asynchronous Receiver/Transmitter (UART) [4] port. IITH-mote can be programmed using TinyOS [5] or assembly language. We use TinyOS platform for programming the motes.

1.2 TinyOS

TinyOS is a free and open source software used for embedded systems. One special feature of TinyOS is, it is a component-based operating system developed for implementing applications of WSNs. We can structure the problem into component implementations and reuse them. This is the optimal way to design and combine the code. NesC (Network Embedded System C) [6], which is a dialect of C language, is the programming language used in this embedded operating system. TinyOS provides various interfaces and components to work with the WSN motes. Unlike other WSN programming platforms, TinyOS supports event based programming and is suitable for low power applications. TinyOS supports a variety of microcontroller families and radio chips. It efficiently manages the limited power available and renders excellent support for wireless networking.

1.3 System model and details

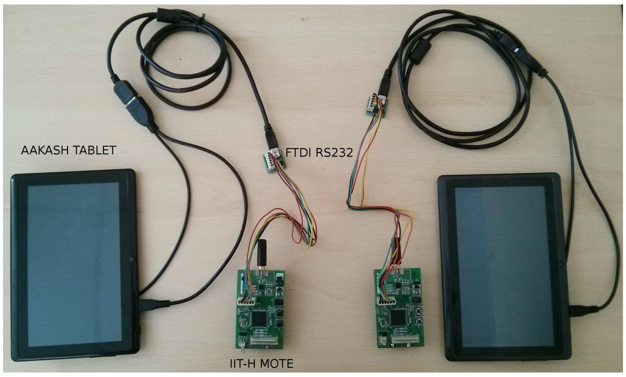

Figure 1.2: Text messaging between two android devices (Aakash tablet) through IIT-H motes

Fig.2 shows peer to peer communication between android devices.Here the android devices used are Aakash tablets [7] The communication between motes is based on the Zigbee standard. The IITH-mote is connected to the android device through microUSB serial port.

1.3.1 Radio communication

Radio communication between the IIT-H motes is done through packets. This packet based communication is completely controlled by the TinyOS application.

TinyOS application

A basic TinyOS application for radio communication contains 4 files.
1. Configuration file

2. Module file
3. Make file
4. Header file

Configuration file contains the components that are used on IIT-H mote. Module file contains variables, code logic and interfaces. Makefile allows platform selection and includes rules required to make the application. Header file contains packet structure which is important. Following is an example of packet structure.

```
#ifndef PACKET_STRUCTURE_H
#define PACKET_STRUCTURE_H
typedef nx_struct AmMsg{
nx_uint8_t counter;
} AmMsg;
#endif /* PACKET_STRUCTURE_H */
```

In the above example, packet contains a single identifier named 'counter' of `nx_uint8_t` data type. Data type `nx_uint8_t` in TinyOS is equivalent to unsigned character data type. The default maximum payload size of a packet in TinyOS is 28 bytes. In this we used maximum payload size.

1.3.2 Serial communication

Serial communication is helpful in communicating with external devices. In general, a WSN will have a gateway for collecting data, observing network traffic and sending commands to the network. A gateway is nothing but a computing device connected with a node of that WSN. Here the communication between node and computing device is serial (UART) communication. TinyOS provides two different types of serial communication. They are packet based and packet less serial communication.

Packet based serial communication

In Packet based serial communication, both computing device and mote communicate using packets. TinyOS platform on computing device provides different tools for packet based serial communication. These tools help in creating, sending packets to the mote, parsing the packet sent by mote and analyzing the data. Listen tool is a basic packet sniffer provided by TinyOS used for packet based serial communication. Let us assume that the packet discussed above in radio communication is sent serially. Listen tool displays that packet as

ff ff 00 00 04 22 06 00 0B

The interpretation of the above data is shown in the Table. 1.1

Dest addr	Link source addr	msg len	group ID	handler ID	counter
ff ff	00 00	04	22	06	0B

Table 1.1: Interpreting packet data

The payload of the packet is counter and all remaining fields are header information. On PC TinyOS provides tools for separating header information from payload

But the disadvantage of this mode is that there are no sophisticated tools in android platform for supporting packet based communication. Hence we use packet less serial communication between IIT-H mote and tablet.

Packet less serial communication

In this mode raw data sent from tablet is converted to serial data by RS232 FTDI chip. It does not follow any packet structure. Every serial communication will have a baud rate and IITH-mote supports a baud rate of 57600. For this purpose an android application called B2B was developed for this. This application is developed based on a demo application 'TN 147 java D2XX' [8] provided by FTDI. B2B will automatically detect the device whenever mote is connected to the tablet. This application has default settings of baud rate, data bits and parity bit to be used. B2B writes the message to the micro usb port byte by byte. RS232 chip receives the bytes and convert them into UART format and sends byte by byte serially to the mote.

1.3.3 Message structure

While creating the WSN each node is given a unique identification called node ID with which user is identified. This node ID is used as source address and destination address while messaging. B2B application provides a text field for writing the message. After completion of writing, touch send button next to the text field to send the message. Text messaging follows a message structure shown in Fig. 1.3

Figure 1.3: Message structure

1.4 Advantages

ZigBee provides low cost and low power connectivity devices where lifetime lasts for months but does not provide bit rate as high as those enabled by Bluetooth. But the bit rate provided by Zigbee is sufficient for text chatting. Zigbee uses an unlicensed globally compatible ISM band ranging from 2.4 GHz to 2.4835 GHz. So a low cost communication system can be developed for use in residential areas. Since the android platform is now interfaced with WSNs, it should be possible to incorporate this in mobile devices on a large scale at low cost.

1.5 Conclusion

In this project we implemented a peer to peer message chatting application in android platform over wireless sensor networks. Since we use low power nodes with IEEE 802.15.4 compliant RF Transceiver for radio communication, the range of a node is limited to few tens of meters. The range of the network can be increased by using multiple hops in between. We can implement different topologies with good networking to cover larger areas. The realization of such a network in residential areas should be the focus of future research.

Chapter 2

Real time half-duplex voice calling over IEEE 802.15.4/Zigbee standard using Android platform

2.1 Introduction

As mentioned in the first chapter, Wireless Sensor Networks(WSNs) are primarily used for Health care monitoring, Industrial monitoring, Environmental sensing, Data logging etc. These applications require low bit rates but IEEE 802.15.4/Zigbee standard supports bit rates of 250 Kbits/s [9], which is much higher than what is required for the above applications. Also, WSN motes consume less power and require minimal maintenance. These unique properties of IEEE 802.15.4/Zigbee standard can be exploited to enable short range voice communication over WSNs. The main application of this can be found in underdeveloped areas that suffer from power outages and poor economy. Such a network was implemented in [10] and tested. They implemented real time voice communication over a Zigbee network which covers an entire village. The main disadvantage of the network was that proprietary handsets were used which were expensive.

As a first step towards developing a short range communication network which uses commercially available devices, we implemented peer to peer half-duplex voice communication over IEEE 802.15.4/Zigbee using cheap open source Android devices. Since Android devices have high speed processors, we can implement the necessary voice codecs for effective voice communication on them. Also, necessary hardware for voice input and output is built into the device, so no other hardware apart from Zigbee motes is required for implementation. We used Aakash tablets running Android, Zigbee based TelsoB [11] motes and TinyOS platform for programming the motes.

2.2 Implementation details

The user handset in this implementation is Aakash tablet interfaced with TelosB mote (See Fig. 2.1) using USB (Universal Serial Bus) On-The-Go (OTG) cable. Aakash tablet uses packet-less serial communication [12] to communicate with the TelosB mote. The TelosB motes communicate through packet based radio communication. Fig. 2.2 shows the flow diagram of the peer to peer voice call. At the sender side, the process

Figure 2.1: User handset (left) and Android application layout (right)

of recording voice, encoding voice data and sending the encoded data happens simultaneously.Encoding and sending data to TelosB mote will be done at intervals of 20 ms. Each encoded data of 20 ms voice is loaded into a packet and sent over radio by the TelosB mote. While at the receiver end, the received packets are parsed and forwarded to the Android device. The data received by the Android device will be decoded and played. For this project, we developed an Android application for communicating with TelosB mote.

2.2.1 User interface layout of Android application

The user interface layout (see Fig. 2.1) of the application contains a text view, a text box and three user buttons. Text view gives the instructions of operation. Text box is to input the destination node id which is given to each TelosB mote. The three buttons are Enter destination, Record and Stop. After writing the destination node id in the text box, click on Enter destination button. After that by clicking on Record button the voice call is initiated. Stop button is used to end the voice call.

2.2.2 Voice Recording

The voice data from the microphone is handled using AudioRecord Android Application Program Interface (API). The main input parameters for this API are recorder sample rate, audio encoding format and buffer size to store the voice data. There are five supported sample rates ranging from 8 Kbps to 44.1 Kbps. Owing to the low bit rate of Zigbee standard, we chose the minimal possible sample rate of 8 Kbps. The audio encoding format used is 16 bit PCM (Pulse Coded Modulation) as 8 bit PCM is not supported by all android devices. The buffer size is 4096 bytes for the above mentioned sampling rate and recording format. The recorded data is written into a temporary file in frames of size 4096 bytes.

2.2.3 Speex Voice Codec

The raw voice data needs moderately high bit rates for establishing real time voice communication. In order to work with low bit rate IEEE 802.15.4 standard voice data should be compressed. For this purpose it is necessary to implement a voice codec on Android platform. There are many voice compression tools

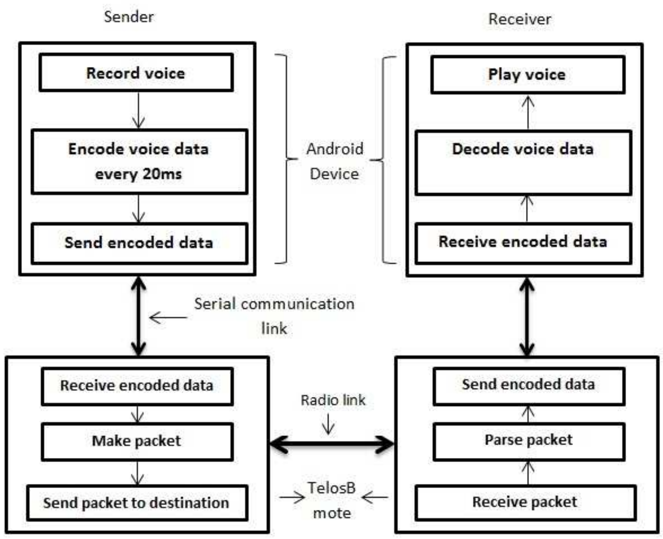

Figure 2.2: Flow diagram of the voice call

available. We chose to implement Speex codec [13] since it is the best open source codec available and has been optimized for high quality speech and low bit rates. Also it is possible to control the trade off made between quality and bit rate by using Speex codec. Unlike many other speech codecs, Speex is not designed for mobile applications but rather for packet networks and VoIP (Voice over IP) applications. So, initially we used Jspeex [14], which is the Java port of the Speex codec, for the implementation. But, encoding and decoding processing time using Jspeex was adding delay and was not suitable for real time application. The alternative method is implementing native C codes of Speex in android which is not straight forward as Android supports only Java based libraries. So, for doing this we used Android NDK (Native Development Kit) tool set which allows to implement parts of an Android application using native code languages such as C and C++. Implementation of Speex codec involves encoding of recorded voice data at sender end and decoding of the received voice data at the receiver end.

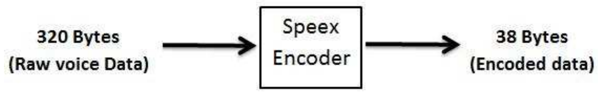

Figure 2.3: Speex encoder

Encoding

Speex codec uses multiple bit rates and supports different sampling rates of 32 KHz (Ultra-Wideband), 16 KHz (Wideband) and 8 KHz (Narrowband). Compression ratio of voice data depends on sampling rate used. In this project, we used Narrowband sampling rate which has the highest compression ratio and offers telephone quality voice. Speex encoder operates in frames. A frame contains raw voice data corresponding to 20 ms of voice. Frame size depends on sampling rate used by encoder and recording format, sampling rate used in voice recording. In Narrowband operation mode and for the parameters used in voice recording section, the frame size is 320 bytes (160 samples). These frames are read from the temporary file (created while recording voice) and given as input to the Speex encoder, which compresses the given frame and returns 38 bytes of encoded voice data. Now the 38 bytes of encoded voice data is written into a file, which will be later used by the Serial communication part (subsection D).

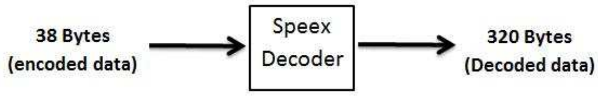

Figure 2.4: Speex decoder

Decoding and playing

At the receiving end, 38 bytes of encoded voice data is received from TelosB. Every time 38 bytes of data is received, it is stored in receive buffer and this buffer is given as input to the decode method, which decodes the data and returns 320 bytes of voice data. Now, the decoded data should be played immediately in order to make the communication real time. So, these 320 bytes are stored in a dynamic buffer and every time after decoding 38 bytes, 320 bytes are appended to this dynamic buffer.

This dynamic buffer is used for playing the voice. AudioTrack Android API is used for this purpose. Write and Play methods of this API are used here. When 320 bytes are decoded for the first time, Play method is initiated and starts playing the voice. Write method is used for writing the data from dynamic buffer to Play method. As the dynamic buffer will be modified every 20 ms, the voice will be played in real time until the sender ends the voice call.

2.2.4 Serial Communication

Android device and TelosB mote communicate using serial communication. TelosB mote has a USB port through which it can be interfaced with external devices. The micro USB port of Android device is connected to the USB port of TelosB mote using OTG cable. The USB port of TelosB mote is internally interfaced with FT232BL [15] chip which is used for conversion of serial data format to USB data format and vice versa. This serial communication works with a baud rate of 115.2 Kbps.

FTDI device manager

The serial communication on the Android device is managed by using FT_device [16] Android API and ftdi D2xx [16] Java library. This API can be used for managing and configuring FTDI [17] devices connected externally to the Android device. The FT232BL chip embedded on the TelosB mote is well supported by this API. The Android application is designed such that the application starts automatically on connecting TelosB mote to Android device.

Sending serial data

FT_device API provides write method to send the encoded data to the microUSB port. To make the communication real time, we need to ensure that voice data is encoded and sent simultaneously. For this purpose, we are using a thread for reading 38 bytes of encoded data from encoded file and writing it to the micro USB port every 20 ms. Along with this 38 bytes of data, an extra byte containing the destination node id is sent before 38 bytes, which will be used by TelosB for further processing.

Receiving serial data

TelosB mote sends data byte by byte to the micro USB port. These bytes are stored into a buffer queue. The getQueueStatus method provided by FT_device API is used to check the length of the buffer. When the length of the buffer reaches 38 bytes, these 38 bytes of data are read using Read method and loaded into a byte array. This byte array will be used by the decoding section.

2.2.5 TinyOS program

TelosB motes communicate through packet based radio communication. For this project, TinyOS v2.1.2 platform is used for programming TelosB motes. In this application, TinyOS program handles Serial communication with Android device and Radio communication between two motes.

Mote to Android device communication

The TinyOS interface used for this serial communication is UartStream which provides Send method for sending data and an event Receivedbyte which signals the receipt of a byte. As and when Android device starts sending data, Receivedbyte event is triggered which stores the received byte into a buffer. When the buffer size reaches 39 bytes (including destination node id), this data will be processed and used to form a packet and is sent over radio. For sending the received data from other mote, Send method is called. Send takes pointer to the stored received data and length of the data to be sent as its arguments.

Mote to mote communication

TelosB uses cc2420 radio for communicating with other motes. cc2420 uses CSMA/CA [18] as default MAC layer. This communication is purely packet based. The structure of a packet can be defined in a header file. The default packet size in TinyOS is 28 bytes. As we need to send 38 bytes of encoded data and destination node id, the packet size has to be 39 bytes. Packet size can be changed by modifying few header files in the TinyOS directory. In this project, the packet contains a variable for destination node id and a byte array of size 38 bytes for storing encoded data.

After receiving every 39 bytes from USB port, a packet will be created. The first byte of the 39 bytes is destination node id which is loaded into the variable of created packet and the remaining 38 bytes are copied into byte array. For sending the packet, we used Send method provided by AMSend interface. Send method can broadcast the packet or it can send it to a particular destination by writing the destination node id in the address field of the method.

For receiving data from other motes, AMReceive interface is used. This interface provides an event called Receive for handling the received packets. This event parses the received packets and extracts the data from it. The extracted 38 bytes of data is sent to Android device using UartStream interface.

2.3 Conclusion

We implemented real time half duplex voice calling on Android devices with IEEE 802.15.4/Zigbee as underlying communication standard. The user handset developed in this project is Android device interfaced with TelosB mote, both are popular and cheaply available in the market. We can extend this peer to peer half duplex voice communication to full duplex voice communication over a network which can cover an entire village (which is proved in [19]). This application can also be extended to support reliable messaging [20] and file transfer as they need not be done in real time. This kind of network is likely to be very useful to the people in underdeveloped areas.

Chapter 3

Non-real time voice messaging over static IEEE 802.15.4/Zigbee network

3.1 Introduction

Real-time full duplex voice communication over IEEE 802.15.4 network is implemented in [10]. Figure 3.1 depicts the Lo3 mesh architecture used in [10]. There are two types of nodes in Lo3 : static nodes which are infrastructure nodes acting as relay nodes for data communication, and user nodes acting as data originators or terminators which can be mobile inside the network. The infrastructure nodes are deployed on rooftops (5-10m height) and a range of over 400m can be achieved for 0dBm transmit power and 8dBi omnidirectional antenna mounted at a height of less than 3m. With this arrangement, an area of about 3km diameter can be covered by a mesh of about 20-30 infrastructure nodes. The user handset (see in 2.1) developed in this project can be used as user node in the network which is described above. This will reduce the cost of the network and is easy to implement.

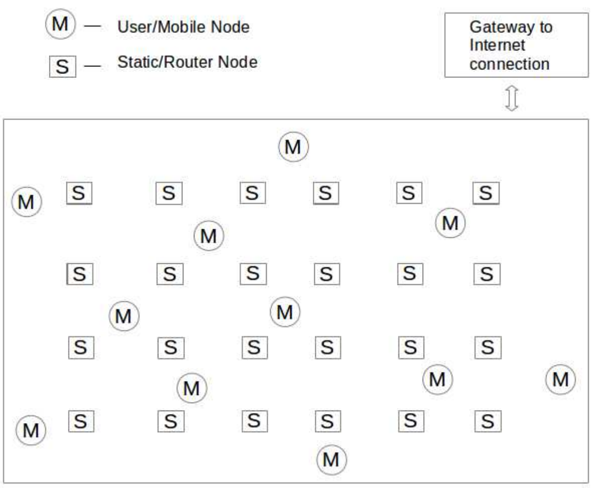

Figure 3.1: LO3 mesh network

3.2 Ideal IEEE 802.15.4 network for real time voice implementation

The default MAC layer used in TelosB mote is CSMA/CA. Before accessing the channel CSMA senses the channel for any activity in the channel. If channel is busy, it backs off for a random period of time and repeats the process again. The CSMA protocol in much of prior work allows for an easy distributed implementation, but it is known to suffer from poor performance in multi-hop networks [21], especially for real-time applications [22]. Self-interference, i.e. simultaneous transmissions across hops of a path, results in capacity reduction and delay unpredictability. In this respect, TDMA is known to perform better. TDMA-based approach also provides good support for duty-cycling and power savings in the network nodes. So in LO3 project, they implemented a TDMA based MAC for the network.

For effective real-time application support, especially for features such as delay-aware scheduling or call-admission control, a centralized approach is intuitively better. A centralized architecture gives significant leverage in effectively addressing the challenges like centralized coordination of time-slots and multiple channels lends itself to a simpler design, as compared to a distributed approach, in a multi-hop network. Although centralized approach is frowned upon for lack of scalability, LO3 MAC showed that the control overheads are not scaling bottlenecks in practice.

802.11b/g (wi-fi standard) has 3 non-overlapping channels of operation, and 802.15.4 has 16 different channels in the 2.4GHz ISM band. Clearly, the use of multiple channels has better potential through- put, if the channel coordination can be done efficiently. Centralized TDMA based approach eases the issue of multichannel coordination significantly. Channel switching does add a small but noticeable overhead, but the benefits of multi-channel operation far outweigh this. LO3 project uses a network of this kind and is very suitable for implementing real time voice application.

3.3 Limiting aspects of User handset for real-time voice

The two limiting aspects of user handset are data rate of serial communication and processing delay at TelosB mote. The data rate of serial communication is 115.2 Kbps. With this set up, the user handset is taking 20 ms time for sending a packet of data which contains encoded data of 20 ms voice. With this processing delay we can implement peer to peer real time voice. Most of the delay is coming from processing delay of TelosB mote. Because, for writing a packet of 38 bytes (Encoded voice data size for 20 ms voice) over serial communication it takes 0.32 ms and the remaining delay is from TelosB mote.

The cause of huge processing delay on TelosB mote is due to low processing speed and CSMA MAC. We have to live with the low processing speed. But, we can reduce the delay due to MAC by changing the medium access layer with TDMA as it is done in LO3 project. LO3 MAC works with TDMA frames, each frame of duration 60 ms. In this 60 ms, a user gets 9 ms of time for sending or receiving the data. In LO3 implementation, a user has to send 45 bytes of encoded voice data in that 9 ms slot, which corresponds to 60 ms of real time voice.

But, the user handset developed in this project is capable of sending 38 bytes in 20 ms which corresponds to 20 ms voice data. The reasons for this are the voice codec and TelosB mote. The voice codec implemented in LO3 user handset works with a sample rate of 5.9 Kbps which generates 45 bytes for 60 ms of voice data. Where as the voice codec implemented in this project works with a sampling rate of 8 Kbps as it is the lowest sampling rate possible on Android platform. The other bottleneck is TelosB mote attached with Android device. In LO3 handset, the encoded data is sent directly from the processor to CC2420 antenna (which is also used by TelosB mote) with out any delay. But here, first the encoded data need to go to TelosB mote, process the data, make packets and sent using antenna which works using CSMA MAC.So, the handset developed can't be used directly in LO3 network. For this to happen, modifications are needed to the frame structure used in LO3.

3.4 Non-real time voice messaging

Before going to implement real time voice communication over network, we tried to implement non-real time voice communication over a network of three infrastructure nodes (See in Figure. 3.2). Users can connect to these infra nodes. This network is static and users are not mobile. Since it is a static network routing is done at the node level. Infra nodes will route the data based on the destination node id. These infra nodes can support any number of users but this network can handle only one active route at a time. Static network architecture can be of any design based on the application. Following is an example network which is deployed and tested in real time conditions. We observed very small amount of jitter while playing the voice message received which is due to packet losses resulted because of bad channel conditions and it can be eliminated by using a reliable TDMA MAC layer.

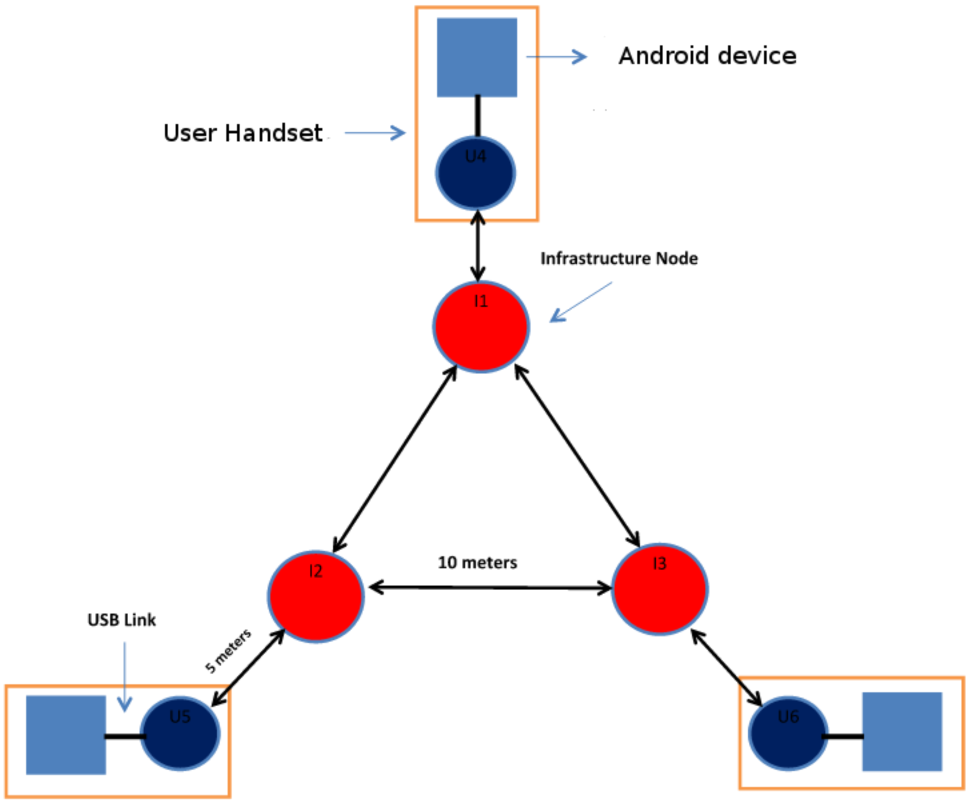

Figure 3.2: Static network

Chapter 4

Impact of multiple limited backhaul capacities on user scheduling in heterogeneous networks

4.1 Introduction

Several rural areas across the globe still suffer from limited and discontinuous power supply. For example, according to a survey done by [23], several villages in India have, on an average of, 15 hours of power outage per day. Several IEEE 802.11 standard based solutions exist for Internet connectivity in diverse scenarios. However, in presence of such limited power supply these solutions are no more applicable. Hence, a low cost and low powered IEEE 802.15.4 based solution for voice services has been provided in [24]. More details regarding comparison of IEEE 802.11 and IEEE 802.15.4 based networks can be found in [24]. However, a low cost and self-powered IEEE 802.15.4 based solution for data services in rural areas is needed. Further, user scheduling in presence of limited backhaul capacity for cellular networks has been examined in [25]. However, in presence of a multi-hop networks with limited backhaul capacity of both Internet gateway (IG) and relay nodes the joint user scheduling with proportinal fairness is an open research problem. This is the motivation for this work.

In [26], they have implemented a connection oriented, multihop, multichannel TDMA based MAC on a village network. They used this network for establishing voice calls among the villagers. Their analysis says, CSMA MAC gives poor performance for bulk data transfer due to intrapath interference which reduces data rates even more which is very critical in our case. The advantage with CSMA is that it is a distributed MAC. So, routing can be done at node level which is connectionless. But it is hard to implement routing protocol on it as there is no standardized routing protocol over IEEE 802.15.4. There is a team which is working on standardizing a routing protocol RPL (Routing Protocol over Low power lossy networks) but it is still under development and also it did not account the mobility of of the nodes in the network which is common in our netwrok. So we adopted the MAC used in [26] with a centralized node for routing the data. The centralized node need not be a power hungry and powerful processor. An IEEE 802.15.4 node can serve the purpose which can be self-powered. Many applications of IEEE 802.15.4 are static networks and uses CSMA as MAC layer. Hence research in the field of mobile IEEE 802.15.4 networks which use TDMA as

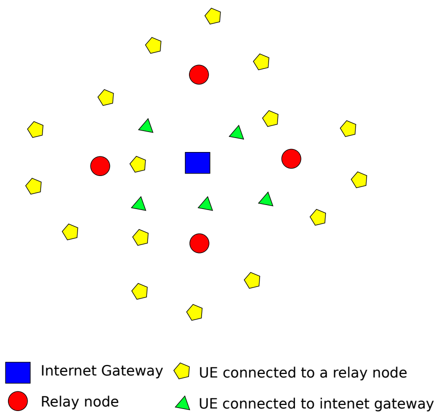

Figure 4.1: *Network model*

MAC layer is little. [24], [26] are good examples of research in the aforementioned field. These two papers describe creating IEEE 802.15.4 multihop network for extending the cellular network range into a village and also establishing voice calls among them. They designed a MAC called PIP (Packets In Pipe) based on TDMA. The MAC frame also includes a field for handling mobility of users in the network.

The multiop network in [24] can cover a small village with the help of 17 nodes. They have achieved a throughput of 60 kbps which shows that even though the maximum data rate of IEEE 802.15.4 channel is 250 kbps, relaying data over multiple hops will reduce the throughput of the network. In our network case the maximum number of nodes in path are two(one hop). So data rates will not reduce much as that in the previous case. But the handsets used in [24] are proprietary hardware. In our case, handsets are android mobile interfaced with IEEE 802.15.4 node also known as mote (See in Fig. 2.1). We have used IIT-H mote for our project. The interfacing of mote with android device is done in [20] which made accesing internet and making phone calls easier.

Given the user association in the network, scheduling of the users is the critical job. So far no research has been done on optimizing the time fractions that are going to be allotted to the users under IEEE 802.15.4 framework. Assuming different link rates are possible with motes then definitely optimization can be done on user scheduling. [27] and [28] have done optimization on user scheduling and user association under infinite backhaul capacity which is not practical in IEEE 802.15.4 network. [25] analyzes impact of single limited backhaul capacity on user scheduling in heterogeneous networks. It did not provide complete solution for for optimizing user scheduling in one of the three cases it studied.

4.2 System Model

We consider a network comprising of one Internet Gateway and W relays and N users (See Figure 4.1). Let I represent Internet Gateway node, R be the set of relay nodes and $\mathcal{N}$ be the set of users. Internet Gateway is connected to internet through a backhaul capacity of C_i. Each relay node $j \in R$ is connected to Internet Gateway through wireless channel with C_j backhaul capacity. Here the user handsets are zigbee mote connected to android mobile (See Figure. 2.1). Even though the maximum data rate possible is 250 kbps but the speed of UART connection acts as bottleneck for data rates. So different zigbee mote manufacturers provide different UART speed.

Hence the link data rates of users might not be same. Let us assume that the Internet Gateway knows the channel capacity of relay node apriori and also relay nodes know the user data rates before hand. Here the backhaul capacities of Internet Gateway and relay nodes are limited. If a user wants to access the internet the data has to go through two limited backhaul channels.

Let λ_i be the throughput of ith user. It was shown in [29] that global proportional fairness can be achieved by maximizing the following objective function $\sum_{i\in\mathcal{N}} \log(\lambda_i)$. R_{ji} be the data rate of ith user connected to jth relay node and α_{ji} be the time fraction allotted to ith user. The optimization problem is finding α_{ji}s which maximize the objective function. S be the scheduling problem to solve, then

$$\mathbf{S} = \max_{\lambda_i, \alpha_{ji}} \sum_{i\in\mathcal{N}} \log(\lambda_i)$$

subject to the conditions:

$$\lambda_i = \sum_{j\in R_s\cup\{I\}} R_{ji}\alpha_{ji} \quad \forall i \in \mathcal{N} \tag{4.1}$$

$$\sum_{i\in\mathcal{N}} R_{ji}\alpha_{ji} \le C_j \quad \forall j \in R_s \tag{4.2}$$

$$\sum_{i\in\mathcal{N}} R_{ji}\alpha_{ji} \le C_I \quad j = I \tag{4.3}$$

$$\sum_{i\in\mathcal{N}} \alpha_{ji} \le 1, \quad \forall j \in R_s \cup \{I\} \tag{4.4}$$

$$\alpha_{ji} \ge 0, \quad \forall i \in \mathcal{N}, \forall j \in R_s \cup \{I\} \tag{4.5}$$

4.1 is the throughput of ith user associated with different relay nodes or Internet Gateway. 4.2 and 4.3 are limitations relay node and Internet Gateway backhaul capacity respectively. 4.4 and 4.5 are constraints on the user scheduling.

As it is discussed in [26], the above problem can be decomposed into $R+1$ independent proportional fair scheduling problems and [26] divides these $R+1$ problems into two categories. First one is Internet Gateway problem and the other one is jth relay node problem. [26] also provides the proof for this decomposition, where optimal schedules for local problems will optimize the global problem. In [26], backhaul capacity of Internet Gateway is infinite. So the solution for the first category becomes equal time scheduling to users. But in our case the backhaul capacity of Internet Gateway is also limited. So the decomposition becomes differnt from [25].

4.2.1 Decomposition

The local problem for Internet Gateway is

$$\mathbf{S}^{\mathbf{I}}_{\mathbf{Local}} = \max_{(\alpha_{Ik})_{k\in A_I}} \sum_{k\in A_I} \log(n_k R_{Ik}\alpha_{Ik})$$

$$\sum_{k\in A_I} n_k R_{Ik}\alpha_{Ik} \le C_I \qquad (\mu)$$

$$\sum_{k\in A_I} \alpha_{Ik} \le 1 \qquad (\beta)$$

$$\alpha_{Ik} \ge 0, \quad \forall k \in A_I \qquad (l_k)$$

Here, n_k is user density of kth link, the number of users get serviced by kth particular link. n_k will be one if kth link is user or will be number of users connected to the relay node if the kth link is a relay node. A_I is the set of users and relay nodes connected to the Internet Gateway. In this case R_{Ik}s may also contain channel capcity of the links. And $\mu \geq 0, \beta \geq 0$, $ł_k \geq 0$ and $\rho_k \geq 0$ are dual variables pertaining to backhaul limitation of Internet Gateway, time fraction constraint, non-negativity of timefraction constraint and user density constraint.

For relay node $j \in R_s$, the local problem with backhaul capacity C_j is

$$\begin{aligned}
\mathbf{S^j_{Local}}(C_j) = \max_{(\alpha_{Ik})_{k\in A_j}} & \sum_{k\in A_j} \log(R_{jk}\alpha_{jk}) \\
& \sum_{k\in A_j} R_{jk}\alpha_{jk} \leq C_j \qquad (\mu_j) \\
& \sum_{k\in A_j} \alpha_{jk} \leq 1 \qquad (\beta_j) \\
& \alpha_{jk} \geq 0, \quad \forall k \in A_j \qquad (l_{jk})
\end{aligned} \tag{4.6}$$

where, $\mu_j \geq 0$, $\beta_j \geq 0$ and $l_{jk} \geq 0$ are dual variables pertaining to backhaul limitation of jth relay node, time fraction constraint and non-negativity of timefraction constraint. A_j is the set of users and relay nodes connected to the jth relay node.

In [26] the solution for the relay node case is obtained by solving the following Lagrangian equation

$$\begin{aligned}
L(\alpha_j;\mu_j,\beta_j,l_j) = & -\sum_{i\in A_j}\log(\alpha_{ji}) + \mu_j(\sum_{i\in A_j} R_{ji}\alpha_{ji} - C_j) \\
& + \beta_j(\sum_{i\in A_j}\alpha_{ji} - 1) - \sum_{i\in A_j} l_{ji}\alpha_{ji}
\end{aligned}$$

The Karush-Kuhn-Tucker (KKT) conditions [30] for the above lagrangian equation are

$$\frac{\partial L}{\partial \alpha_{ji}} = 0 \Longrightarrow \alpha_{ji} = \frac{1}{\mu_j R_{ji} + \beta_j - l_{ji}} \ \forall i \in A_j$$

$$\beta_j(\sum_{i\in A_j}\alpha_{ji} - 1) = 0 \tag{4.7}$$

$$\mu_j(\sum_{i\in A_j} R_{ji}\alpha_{ji} - C_j) = 0 \tag{4.8}$$

$$l_{ji}\alpha_{ji} = 0, \quad \forall i \in A_j \tag{4.9}$$

So the optimal solution will satisfy all the constarints 4.7,4.8 and 4.9. In equation 4.9 we know that $\alpha_{ji} > 0$ so l_{ji} will be zero for all $i \in A_j$. So the time fraction equation becomes

$$\alpha_{ji} = \frac{1}{\mu_j R_{ji} + \beta_j - l_{ji}} \ \forall i \in A_j \tag{4.10}$$

Here we know that both $\mu_j, \beta_j \geq 0$ and α_{ji} has to be greater than zero, so [26] divided the solution for α_{ji}s into three cases. First one is $(\mu_j = 0, \beta_j > 0)$, second case is $(\mu_j > 0, \beta_j = 0)$ and the last one is $(\mu_j > 0, \beta_j > 0)$. The solutions for the first two cases are derived in [26]. They are

4.2.2 case 1 ($\mu_j = 0, \beta_j > 0$)

In this case $\alpha_{ji} = \dfrac{1}{\beta_j}$, $\forall i \in A_j$. These α_{ji}s should satisfy condition 4.7 also. From this condition we will get $C_j \geq \dfrac{\sum_{i \in A_j} R_{ji}}{|A_j|}$. So $\alpha_{ji} = \dfrac{1}{\beta_j}$ is the optimum user scheduling when the backhaul capacity $C_j \geq \dfrac{\sum_{i \in A_j} R_{ji}}{|A_j|}$.

4.2.3 case 2 ($\mu_j > 0, \beta_j = 0$)

Now, in this case $\alpha_{ji} = \dfrac{1}{\mu_j R_{ji}}$, $\forall i \in A_j$. Substituting this in condition 4.7 we will get $\mu_j = \dfrac{|A_j|}{C_j}$. So α_{ji} becomes

$$\alpha_{ji} = \frac{C_j}{|A_j| R_{ji}} \; \forall i \in A_j$$

These α_{ji}s should satisfy condition 4.8. From this condition we will get $C_j \leq \dfrac{|A_j|}{\sum_{i \in A_j} \frac{1}{R_{ji}}}$. This particular user scheduling, $\alpha_{ji} = \dfrac{C_j}{|A_j| R_{ji}} \; \forall i \in A_j$ is valid only for backhaul capacity $C_j \leq \dfrac{|A_j|}{\sum_{i \in A_j} \frac{1}{R_{ji}}}$.

4.2.4 case 3 ($\mu_j > 0, \beta_j > 0$)

The solutions for case 1 and case 2 are well provided in [26]. But for case 3 solution is incomplete. They have provided proof for solution existance in this case and proves the other solutions are invalid for this case. They said solution can be obtained by solving the equations (20) and (21) for $\mu_j \; and \beta_j$. Equations (20) and (21) are obtained by substituting 4.10 into 4.7 and 4.8 respectively.

$$\sum_{i \in A_j} \frac{R_{ji}}{\mu_j R_{ji} + \beta_j} = C_j \tag{4.11}$$

$$\sum_{i \in A_j} \frac{1}{\mu_j R_{ji} + \beta_j} = 1 \tag{4.12}$$

Let us go through the complete solution of this case. We can obtain a relation between μ_j &β_j by dividing 4.11 into partial fractions. Equation 4.11 can be expressed as

$$\frac{1}{\mu_j} \sum_{i \in A_j} 1 - \frac{\beta_j}{\mu_j} \sum_{i \in A_j} \frac{1}{\mu_j R_{ji} + \beta_j} = C_j \tag{4.13}$$

Substituting 4.12 into 4.13 we will get

$$\mu_j C_j + \beta_j = |A_j| \tag{4.14}$$

If we can set the value of either μ_j or β_j the other one can be obtained from 4.14. Substitute β_j value in 4.12 for getting the value of μ_j. So, 4.12 becomes

$$\sum_{i \in A_j} \frac{1}{\mu_j (R_{ji} - C_j) + |A_j|} = 1 \tag{4.15}$$

As i ranges from 1 to $|A_j|$, 4.15 is a polynomial of degree $|A_j|$. So there exists $|A_j|$ solutions for μ_j & β_j

In our network case, we put a minimum lower limit to the link rate for having the resonable performance. Taking UART speed and provision of minimum internet speed for user into consideration we need a minimum link rate $R_{min} = 60$ Kbps. So $\alpha_{ji} R_{ji} \geq R_{min}$ which implies

$$\sum_{i \in A_j} \alpha_{ji} R_{ji} \geq |A_j| R_{min} \tag{4.16}$$

From 4.6 and 4.16 we can write

$$\begin{aligned} |A_j| R_{min} &\leq C_j \\ |A_j| &\leq \frac{C_j}{R_{min}} \end{aligned}$$

As our network is IEEE 802.15.4 based , the backhaul capacity of relay node is $C_j = 250$ Kbps. So $|A_j| \leq 4.16$, which means the maximum number of users can be served by relay node can be $|A_j|_{max} = 4$. Let us look obtain the user scheduling for different possible $|A_j|$ values.

For $|\mathbf{A_j}| = \mathbf{4}$:

Equation 4.15 becomes

$$\sum_{i=1}^{4} \frac{1}{\mu_j (R_{ji} - C_j) + |A_j|} = 1 \tag{4.17}$$

Let

$$\begin{aligned} x &= \mu_j \\ a_i &= (R_{ji} - C_j) \\ c &= |A_j| \end{aligned}$$

Equation 4.17 transforms into

$$\sum_{i=1}^{4} \frac{1}{a_i x + c} = 1$$

In polynomial form, it reduces into

$$\begin{aligned} \left(\prod_{i=1}^{4} a_i\right) x^4 + \left((c-1) \sum_{i=1}^{4} \sum_{\substack{j=1 \\ i<j<k}}^{4} \sum_{k=1}^{4} a_i a_j a_k\right) x^3 + \\ \left(c(c-2) \sum_{i=1}^{4} \sum_{\substack{j=1 \\ i<j}}^{4} a_i a_j\right) x^2 + \\ \left(c^2 (c-3) \sum_{i=1}^{4} a_i\right) x + \left(c^3 (c-4)\right) = 0 \end{aligned}$$

Let

$$
\begin{aligned}
a &= \prod_{i=1}^{4} a_i \\
b &= (c-1)\sum_{i=1}^{4}\sum_{\substack{j=1\\ i<j<k}}^{4}\sum_{k=1}^{4} a_i a_j a_k \\
c &= c(c-2)\sum_{i=1}^{4}\sum_{\substack{j=1\\ i<j}}^{4} a_i a_j \\
d &= c^2(c-3)\sum_{i=1}^{4} a_i \\
e &= c^3(c-4)
\end{aligned}
$$

The roots are:

$$
\begin{aligned}
x &= -\frac{b}{4a} - \frac{p_4}{2} - \frac{\sqrt{p_5 - p_6}}{2} \\
\text{or } x &= -\frac{b}{4a} - \frac{p_4}{2} + \frac{\sqrt{p_5 - p_6}}{2} \\
\text{or } x &= -\frac{b}{4a} + \frac{p_4}{2} - \frac{\sqrt{p_5 + p_6}}{2} \\
\text{or } x &= -\frac{b}{4a} + \frac{p_4}{2} + \frac{\sqrt{p_5 + p_6}}{2}
\end{aligned}
$$

where

$$
\begin{aligned}
p_1 &= 2c^3 - 9bcd + 27ad^2 + 27b^2e - 72ace \\
p_2 &= p_1 + \sqrt{-4\left(c^2 - 3bd + 12ae\right)^3 + p_1^2} \\
p_3 &= \frac{c^2 - 3bd + 12ae}{3a\sqrt[3]{\frac{p_2}{2}}} + \frac{\sqrt[3]{\frac{p_2}{2}}}{3a} \\
p_4 &= \sqrt{\frac{b^2}{4a^2} - \frac{2c}{3a} + p_3} \\
p_5 &= \frac{b^2}{2a^2} - \frac{4c}{3a} - p_3 \\
p_6 &= \frac{-\frac{b^3}{a^3} + \frac{4bc}{a^2} - \frac{8d}{a}}{4p_4}
\end{aligned}
$$

For $|A_j| = 3$:
The equation becomes

$$
\sum_{i=1}^{3} \frac{1}{a_i x + c} = 1
$$

In polynomial form, it reduces into

$$\left(\prod_{i=1}^{3} a_i\right) x^3 + \left((c-1)\sum_{i=1}^{3}\sum_{\substack{j=1 \\ i<j}}^{3} a_i a_j\right) x^2 + \left(c(c-2)\sum_{i=1}^{3} a_i\right) x + \left(c^2(c-3)\right) = 0$$

Let

$$a = \prod_{i=1}^{3} a_i$$
$$b = (c-1)\sum_{i=1}^{3}\sum_{\substack{j=1 \\ i<j}}^{3} a_i a_j$$
$$c = c(c-2)\sum_{i=1}^{3} a_i$$
$$d = c^2(c-3)$$

the general formula for the roots, in terms of the coefficients, is

$$x_k = -\frac{1}{3a}\left(b + u_k C + \frac{\Delta_0}{u_k C}\right), \qquad k \in \{1,2,3\}$$

where

$$u_1 = 1, \qquad u_2 = \frac{-1 + i\sqrt{3}}{2}, \qquad u_3 = \frac{-1 - i\sqrt{3}}{2}$$

are the three cube roots of unity, and where

$$C = \sqrt[3]{\frac{\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}}$$

with

$$\Delta_0 = b^2 - 3ac$$
$$\Delta_1 = 2b^3 - 9abc + 27a^2 d$$

and

$$\Delta_1^2 - 4\Delta_0^3 = -27a^2\Delta$$

where Δ is the discriminant discussed above.

For $|\mathbf{A_j}| = 2$:

The equation becomes

$$\sum_{i=1}^{2} \frac{1}{a_i x + c} = 1$$

In polynomial form, it reduces into

$$\begin{aligned}\left(\prod_{i=1}^{2} a_i\right) x^2 + \left((c-1)\sum_{i=1}^{2} a_i\right) x + \\ c(c-2) = 0\end{aligned}$$

$$\begin{aligned}a &= \prod_{i=1}^{2} a_i \\ b &= (c-1)\sum_{i=1}^{2} a_i \\ c &= c(c-2)\end{aligned}$$

The possible roots are,

$$\begin{aligned}x &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} \\ \text{or } x &= \frac{-b + \sqrt{b^2 - 4ac}}{2a}\end{aligned}$$

By substituting the roots obtained in the all possible $|A_j|$'s in 4.15 we can get the corresponding user schedulings α_{ji}'s. And the solution of the Internet Gateway local problem is obtained by solving the following Lagrangian equation

$$\begin{aligned}L^I(\alpha_j; \mu, \beta, l_k, \rho) = -\sum_{k \in A_I} \log(\alpha_{Ik}) + \mu(\sum_{k \in A_I} n_k R_{Ik} \alpha_{Ik} - C_I) \\ + \beta(\sum_{k \in A_I} n_k \alpha_{Ik} - 1) - \sum_{k \in A_I} l_k \alpha_{Ik} - \sum_{k \in A_I} \rho_k n_k\end{aligned}$$

The KKT conditions for the above lagrangian function are

$$\begin{aligned}\frac{\partial L^I}{\partial \alpha_{Ik}} = 0 \implies \alpha_{Ik} = \frac{1}{n_k(\mu R_{Ik} + \beta) - l_k} \ \forall k \in A_I \\ \beta(\sum_{k \in A_I} \alpha_{Ik} - 1) = 0 \\ \mu(\sum_{k \in A_I} R_{Ik} \alpha_{Ik} - C_I) = 0 \\ l_{Ik} \alpha_{Ik} = 0, \quad \forall k \in A_I\end{aligned} \tag{4.18}$$

In 4.18, α_{Ik} is always greater than zero. So we can make l_k equal to zero and

$$\alpha_{Ik} = \frac{1}{n_k(\mu R_{Ik} + \beta)} \; \forall k \in A_I$$

The solution for different cases can be obtained using the aforementioned procedures.

4.3 Conclusion

We analysed the effect of multiple limited backhaul capacities on user scheduling in the proposed multi-hop network. We provided the complete solution for user scheduling under proportional fair scheme with two limited backhaul capacites in series. This optimization of scheduling will be helpful for the networks using TDMA as the medium access layer.

References

[1] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki. Voice over Sensor Networks. *Real-Time Systems Symposium* .

[2] V. Gabale, B. Raman, K. Chebrolu, and P. Kulkarni. LiT MAC: Addressing The Challenges of Effective Voice Communication in a Low Cost, Low Power Wireless Mesh Network. *ACM DEV10* .

[3] J. T. Adams. An Introduction to IEEE STD 802.15.4. *Aerospace conference* .

[4] http://tutorial.cytron.com.my/2012/02/16/uart-universal-asynchronous-receiver-and-transmitter/.

[5] P. Levis and D. Gay. TinyOS Programming. 1st edition. Cambridge University Press, 2009.

[6] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesc language: A holistic approach to networked embedded systems .

[7] http://www.akashtablet.com/akashview.php.

[8] http://www.ftdichip.com/Support/SoftwareExamples/Android _JAVA_D2XX_Projects.htm.

[9] http://www.ieee802.org/15/pub/TG4.html.

[10] V. Gabale, B. Raman, K. Chebrolu, and P. Kulkarni. LiT MAC: Addressing The Challenges of Effective Voice Communication in a Low Cost, Low Power Wireless Mesh Network. *ACM DEV10* .

[11] http://www.willow.co.uk/TelosB_Datasheet.pdf.

[12] http://www.robotroom.com/Asynchronous-Serial-Communication-1.html.

[13] http://www.speex.org/.

[14] http://jspeex.sourceforge.net/.

[15] http://www.ftdichip.com/Support/Documents/DataSheets/ICs/ DS_FT232BL_BQ.pdf.

[16] http://www.ftdichip.com/Support/Documents/AppNotes/ AN_233_Java_D2xx_for_Android_API_User_Manual.pdf.

[17] http://www.ftdichip.com/Support/SoftwareExamples/Android_JAVA_D2XX_Projects.htm.

[18] F. Wang, D. Li, and Y. Zhao. Analysis of CSMA/CA in IEEE 802.15.4. *IET Communications* 5, (2011) 21872195.

[19] http://www.cse.iitb.ac.in/ vijaygabale/index.php?n=Main.LO3.

[20] B. S. C. T. Reddy, N. S. Teja, and G. V. V. Sharma. Android controlled Zigbee motes for Wireless Sensor Networks. *MASS'14* .

[21] B. Raman and K. Chebrolu. Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks. *11th Annual International Conference on Mobile Computing and Networking (MOBICOM)* .

[22] A. Rowe, R. Mangharam, and R. Rajkumar. RT-Link: A Time-Synchronized Link Protocol for Energy Constrained Multihop Wireless Networks. *SECON* .

[23] http://cgpl.iisc.ernet.in/site/Portals/0/Publications/Presentations/EGM/9_villageelectrificationthroughbiomassgasifier-egm.pdf.

[24] K. C. Vijay Gabale, Bhaskaran Raman and P. Kulkarni. LiT MAC: Addressing The Challenges of Effective Voice Communication in a Low Cost, Low Power Wireless Mesh Network. *ACM DEV10* .

[25] J. Ghimire and C. Rosenberg. Impact of Limited Backhaul Capacity on User Scheduling in Heterogeneous Networks. *IEEE WCNC'14* .

[26] S. B. Bhaskaran Raman, Kameswari Chebrolu and V. Gabale. PIP: A Connection-Oriented, Multi-Hop, Multi-Channel TDMA-based MAC for High Throughput Bulk Transfer. *SenSys10* .

[27] D. Fooladivanda and C. Rosenberg. Joint resource allocation and user association for heterogeneous wireless cellular networks. *IEEE Trans. Wireless Commun.* 12, (2013) 248257.

[28] T. Bu, L. Li, and R. Ramjee. Generalized proportional fair scheduling in third generation wireless data networks. *IEEE INFOCOM* .

[29] F. Kelly. Charging and rate control for elastic traffic. *European transactions on Telecommunications* 8, (1997) 33–37.

[30] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge Univ Pr, 2004.