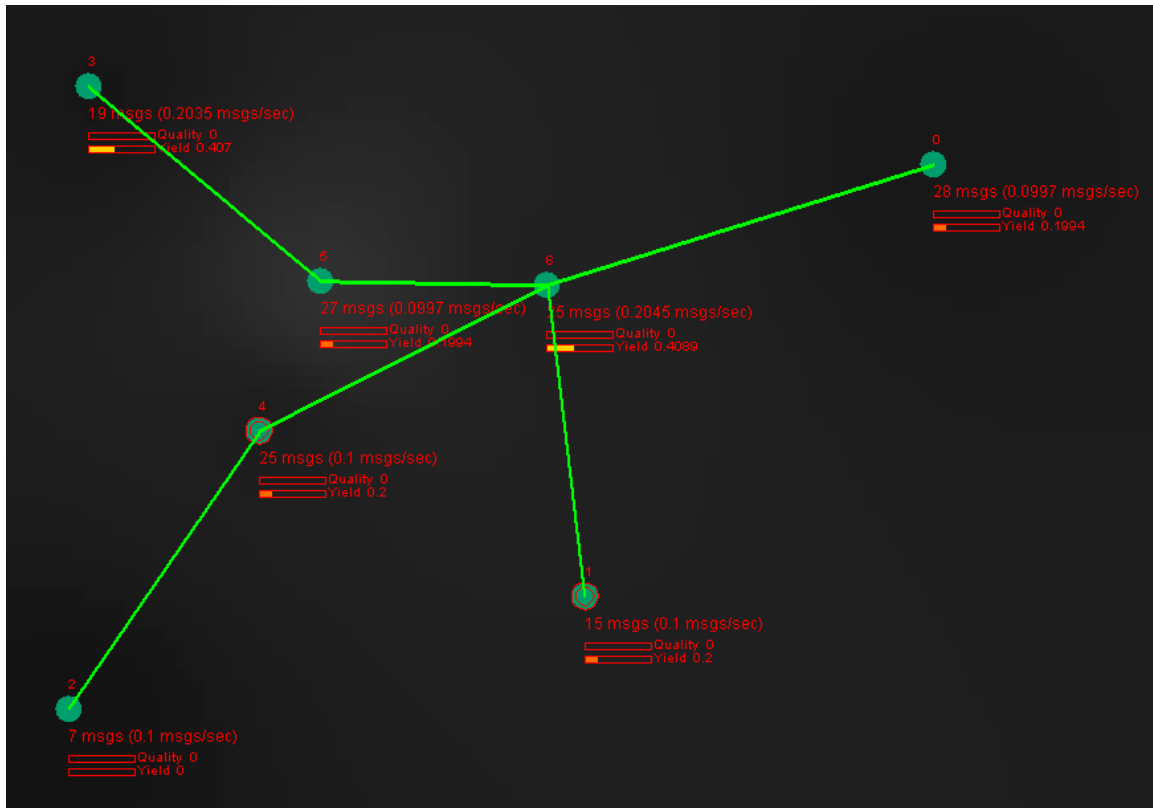


NETWORK MODEL AND DATA COLLECTION

A total of 7 nodes were used to collect the light readings. They were placed in different rooms of the MSCS building in OSU Stillwater. The network was static (no node movement). Due to the routing protocol used, the connectivity between the nodes was dynamic. The node to which a packet is forwarded depends on the existing link quality between the sender and the nodes under consideration. Below is the network topology at one instance.



Each node was programmed to collect and send light readings every 10 seconds. The data was collected for 21 hours.

INTERPRETING THE PACKETS

Consider the first packet in the light data file attached.

```
14 00 00 00 00 00 7E 00 11 7D 00 00 00 00 00 00 0E 00 00 00 00 00 E0 00  
7E 00 2C 00 00 00
```

The hexadecimal values highlighted in Green follow the 'MultihopMsg' structure format given below.

```
typedef struct MultihopMsg {  
    uint16_t sourceaddr; //node that forwarded the packet  
    uint16_t originaddr; //node that actually created the packet
```

```

    int16_t seqno;
    int16_t originseqno;
    uint16_t hopcount;
    uint8_t data[(TOSH_DATA_LENGTH - 10)];
} TOS_MHopMsg;

```

sourceaddr - It is the address from which we received the packet.
originaddr - This is address of the node which actually observed the light readings and created the packet.
hopcount - This is the hopcount of the packet (the number of nodes the packet traveled to reach the source node).
data - This is the actual data sent by the origin node. The data follows the structure SurgeMsg as described below.

So in the packet we are considering,

00 00 00 00 00 00 0E 00 00 00

Source Address = 00 00 (0) (Node we received the packet from)
 Origin Address = 00 00 (0) (Node that observed the light reading)
 Sequence Number = 00 00 (0)
 Original Sequence Number = 00 0E
 Hop Count = 00 00

The packet data highlighted in Yellow is the actual 'data' field in the MultihopMsg structure. It follows the format of the structure 'SurgeMsg' given below.

```

typedef struct SurgeMsg {
    uint16_t type;
    uint16_t reading; //the light reading observed
    uint16_t parentaddr; //node the packet is forwarded to
    uint32_t seq_no;
} SurgeMsg;

```

type - This is the message type used for different kinds of messages exchanged between the nodes. It is 0 for messages containing readings.
reading - The voltage observed on the photo diode.
parentaddr - The node address which the origin node chose to forward the packet to. Node 0 is the base station and it forwards packets to the address 126 which is the address of the UART.
seq_no - The sequence number set by the origin node.

In the packet we are considering,

00 00 E0 00 7E 00 2C 00 00 00

Message Type = 00 00
 Light Reading = 00 E0
 Parent Address = 00 7E (126 which is the address of the UART)
 Sequence Number = 00 00 00 2C

The data highlighted in Red has the following message format.

Length - 1 byte

Frame Control Field – 2 bytes
Data Sequence Number – 1 byte
Destination PAN address – 2 bytes
Destination address – 2 bytes
Active Message ID – 1 byte
Group ID – 1 byte

14 00 00 00 00 00 7E 00 11 7D

In the above data from the packet we are considering,

Length – 0x14

Frame Control Field – 0x0000

Data Sequence Number – 0x00

Destination PAN address – 0x00

Destination address – 0x007E

Active Message ID – 0x11

Group ID – 0x7D

NOTE: When two bytes are to be accounted for a particular field, the byte displayed first is the lower order byte, and the one displayed next is the higher order byte.