NDN-Lite Over LoRa

Zhaoning Kong, Yufeng Zhang, Bo Chen (UCLA)
Sanjeeve (FIU)
Kent (ASTRI)

1 Background

LoRa™

The first low-cost implementation of chirp spread spectrum (CSS) for commercial usage.

- **long distance,**
- **low power consumption**

Application:

Security System

Irrigation System

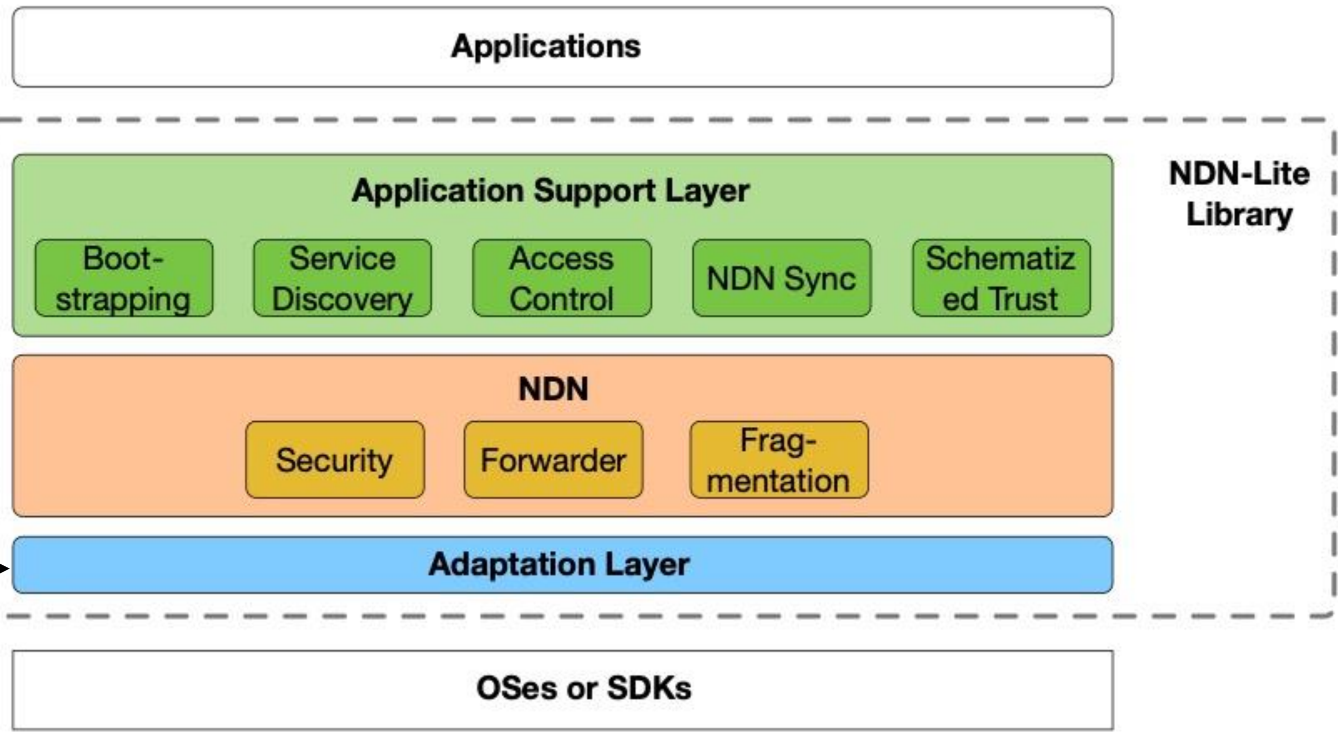
Industrial Monitoring & Control

Automated Meter Readin

1 Design goal

LoRa™

ndn_lora_face_t () →



Applications

NDN-Lite Library

Application Support Layer

Boot-strapping

Service Discovery

Access Control

NDN Sync

Schematized Trust

NDN

Security

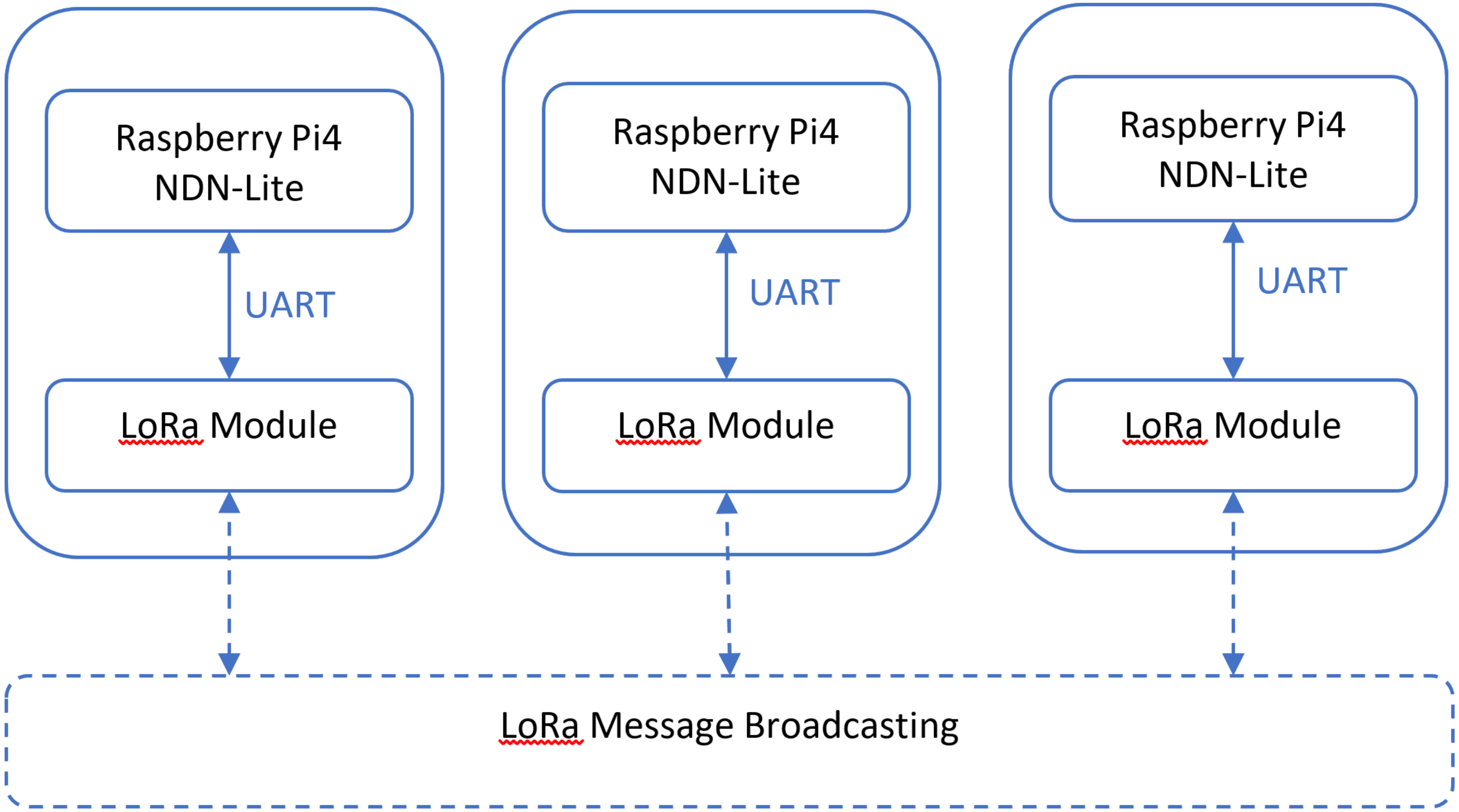
Forwarder

Frag-mentation

Adaptation Layer

OSes or SDKs

3 Architecture



Sending flow:

1) Encode data and use Lora face to send NDN pakcets
2) Raspberry transmits the packet to Lora module via UART
3) Lora module sends out the Lora message

Receiving flow:

1) Lora module receives the lora message from other nodes
2) Use UART to transmit the message to Raspberry
3) Decode the message as the NDN packet

4 Implementation-Software

Flag for detecting a full packet:

```
static uint8_t magic[4] = {0x80, 0xdb, 0xa9, 0x3e};
```

```
ndn_lora_face_send(ndn_face_intf_t* self, const uint8_t*
packet, uint32_t size){
  ndn_lora_face_t* ptr = (ndn_lora_face_t*)self;
  ssize_t ret = write (ptr->fd, packet, size);
  ret += write(ptr->fd, &magic, sizeof(magic));
  if(ret != size + 4){
    return NDN_LORA_FACE_SOCKET_ERROR;
  }
  else{
    return NDN_SUCCESS;
  }
}
```

```
ndn_lora_face_recv(void *self, size_t param_len, void *param){
 ssize_t size;
 int ret;
 ndn_lora_face_t* ptr = (ndn_lora_face_t*)self;
while(true){
   size = recvfrom_lora(ptr);
   if(size > 0){
     ret = ndn_forwarder_receive(&ptr->intf, ptr->buf, size);
    }
   else if(size == 0){ break; }
   else{
      ndn_face_down(&ptr->intf); return; }
  }
  ptr->process_event = ndn_msgqueue_post(self,
ndn_lora_face_recv, param_len, param);
}
```

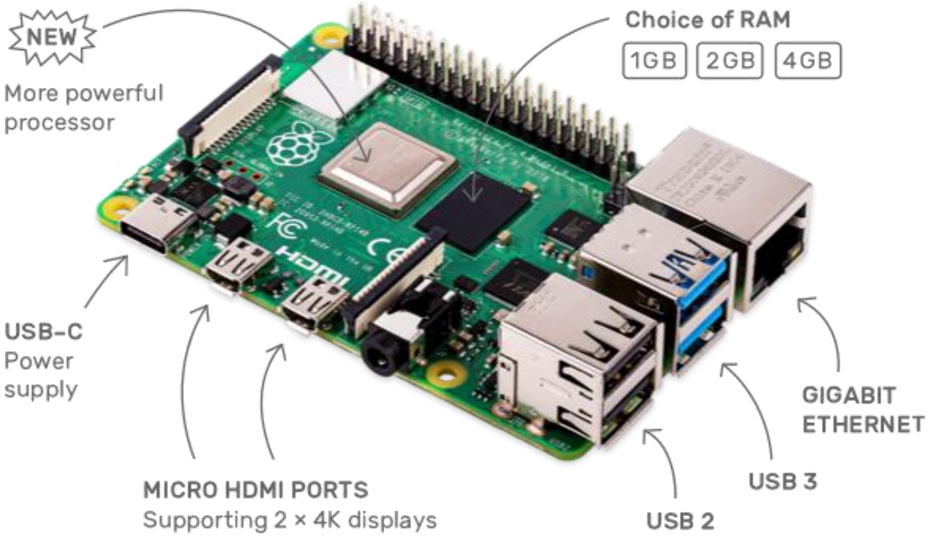
4 Implementation-Software

How to detect a full packet:

```
static ssize_t
recvfrom_lora (ndn_lora_face_t* ptr) {
int buffPos = 0;
uint32_t window = 0;
while (true) {
  if (serialDataAvail(ptr->fd) == 0)
    continue;
  ptr->buf[buffPos] = serialGetchar(ptr->fd);
  window = byte_shift_left(window, ptr->buf[buffPos++]);
  if (memcmp(&window, magic, sizeof(magic)) == 0) {
    return buffPos – 4;
  }
}
```

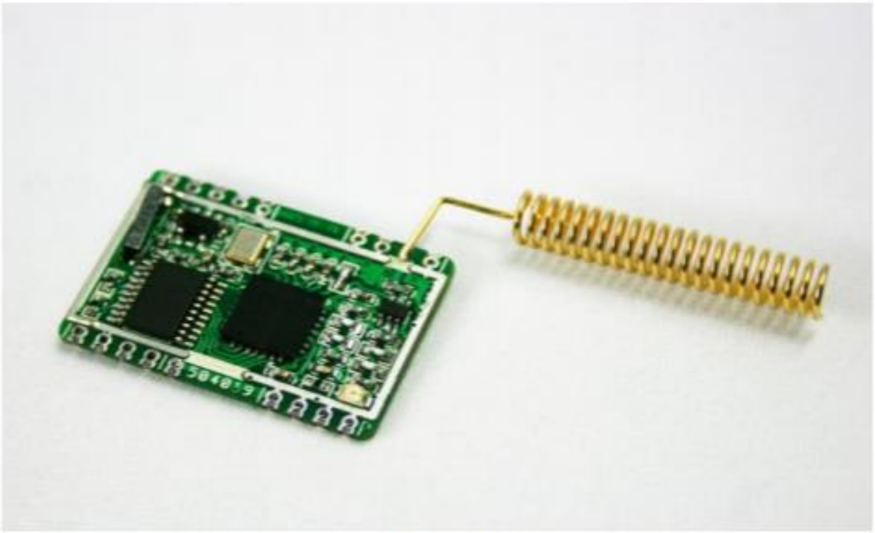
5 Implementation-hardware

NDN lite running on Raspberry pi



Raspberry pi 4:
OS: Raspbian (4.19)
Memory: 4GB
CPU: ARM v8 1.5Ghz

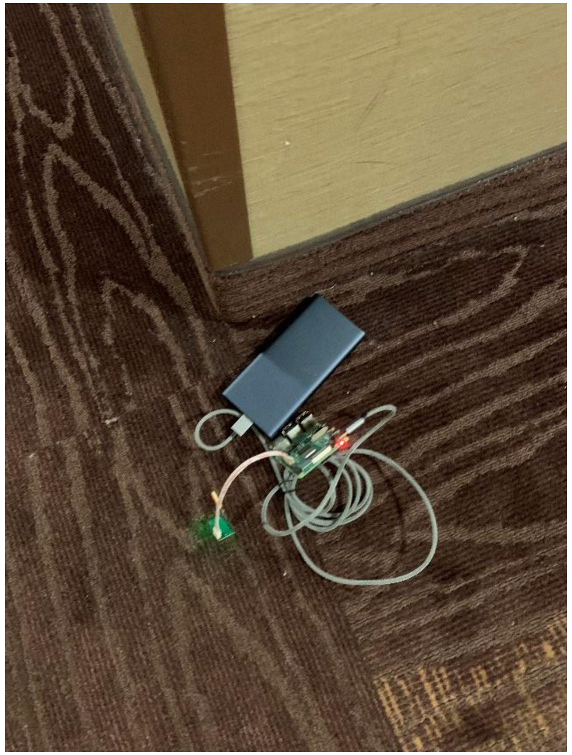
Lora module connected with Raspberry pi



GC-TS12

GC-TS12
Programmable bit rates up to 300 kbps
High sensitivity: down to -148 dBm
Long transmit distance, up to 3000 meters in open area
Low power consumption, 3uA stand-by, 12mA in receiving mode

6 Demo (live show)



One node in the hotel

One node in the outside (McDonald’s)