Develop a “smart” wildlife asset tracking system

Kalonji Bankole
Developer Advocate
Problem statement

- Poachers attack thousands of endangered animals every year

- Researchers have found that animals move differently based on the type of perceived threat.

- Goal is to provide an early warning system to notify rangers when poachers are in the area
Implementation

- Wireless network
- Wearable sensors
- Cloud Analytics services
High Level Architecture

Gateway

End node 1

Sensor
LoRa Radio

End node 2

Sensor
LoRa Radio

Cloud Services
LoRaWAN Gateway

- Raspberry Pi
- LoRaWAN Concentrator Board
- 915 MHz antennae
- PCB Shield
LoRaWAN End Node (Base)
GPS Module (NEO-6M)
LoRaWAN End Node (GPS)
LoRaWAN End Node (GPS)
GPS Raw Output

Example NMEA Output

$VWVLW,0.0,N,0.0,N*4C
$YXMTW,24.0,C*14
$SDDPT,0.5,*7C
$SDDBT,1.8,f,0.5,M,0.3,F*09
$VWVHW,,T,,M,0.00,N,0.00,K*54
$VWVLW,0.0,N,0.0,N*4C
$YXMTW,24.0,C*14
$SDDPT,0.5,*7C
$SDDBT,1.8,f,0.5,M,0.3,F*09
$VWVHW,,T,,M,0.00,N,0.00,K*54
$VWVLW,0.0,N,0.0,N*4C
$YXMTW,24.0,C*14
$SDDPT,0.5,*7C
$SDDBT,1.8,f,0.5,M,0.3,F*09
$VWVHW,,T,,M,0.00,N,0.00,K*54
NeoGPS (Code Snippet)

```c
#include <NMEAGPS.h>  // Load NeoGPS library
static NMEAGPS gps;  // This parses the GPS characters
gps_fix fix;  // This holds on to the latest values

void loop()
{
  while (gps.available(gps_port)) {
    fix = gps.read();
    Serial.print(fix.latitude());
    Serial.println(fix.longitude());
  }
}
```
LoRaWAN End Node (GPS + Accelerometer)
End Node Collar

Animal Sensors
- GPS
- Accelerometer
- Magnetometer
- Temperature
- ..others

Photos from @markusvankempen – MVN, IBM, and Wageningen University
Field Deployment

Animal with tagged collar → LoRaWAN™ → Raspberry Pi Gateway → MQTT → IBM Cloud
Sample App Architecture
Sample App

Add Node

ID:
node1

Longitude:
34.010047

Latitude:
-118.317572

Timestamp:
07/19/2018 03:41:12

Sensor Type:
sound

Sensor Value:
40

Create  Cancel
Sample App

```javascript
var initTrackableAsset = function(id, long, lat, time, type = undefined, val = undefined) {
  console.log("initializing asset: " + id)
  var node = {
    marker: L.marker([long, lat]).addTo(mymap).bindPopup("<b>" + id +"</b>").openPopup(),
    points: [ new L.LatLng(long, lat) ],
    timestamps: [ time ]
  }
  if (type) {
    var node1 = Object.assign({},
      circle: L.circle([long, lat], val, {
        color: 'red',
        fillColor: '#f03',
        fillOpacity: 0.1
      }).addTo(mymap).bindPopup("LoRA Node: " + id)
    }, node)
    assets[id] = node1
```
Sample App

```bash
mqtt_pub -v -i "a:{$IOT_ORG_ID}:client_pub1" -u "{$IOT_API_KEY}" -P "{$IOT_AUTH_TOKEN}" -h "d": {
  "node_id": "node2",
  "lat": "-118.317392",
  "long": "34.100057",
  "timestamp": "2018-06-30T07:10:55.174Z",
  "sensor": {
    "sound": "72"
  }
}
```
Sample App

```javascript
var initMQTTClient = function(mqttCreds) {
  var watson_channel = 'iot-2/type/' + mqttCreds.IOT_DEVICE_TYPE + '/id/' + mqttCreds.IOT_DEVICE_ID;
  var cleanSession = true;

  var subscribeOptions = {
    onSuccess: function() {
      console.log("subscription set")
      mqttClient.onMessageArrived = function (messageObj) {
        var message = JSON.parse(messageObj.payloadString);
        console.log( message);
        if ( ! assets[message['node_id']] ) {
          initTrackableAsset(message['node_id'], message['long'], message['lat'] )
        }
        if (message['sensor']) {
          updateTrackableAsset(message['node_id'], message['long'], message['lat'])
        }
      }
    }
  }
}
```
Sample App
Sample App
Sample App
Sample App
Sample App

```javascript
var drawAssetPath = function( id ) {
    assets[id]['path'] = new L.Polyline( assets[id]['points'], {
        color: "rgb(\" + genRandomNums() + ")",
        weight: 1,
        opacity: 0.8,
        smoothFactor: 0
    });
    assets[id]['path'].addTo(mymap);
};
```
References

https://github.com/ibm/iot-mapping

https://www.slideshare.net/MarkusVanKempen/animal-tracking-using-watson-iot-and-lorawan

https://www.wur.nl/en/project/SmartParks-using-technology-for-anti-poaching.htm