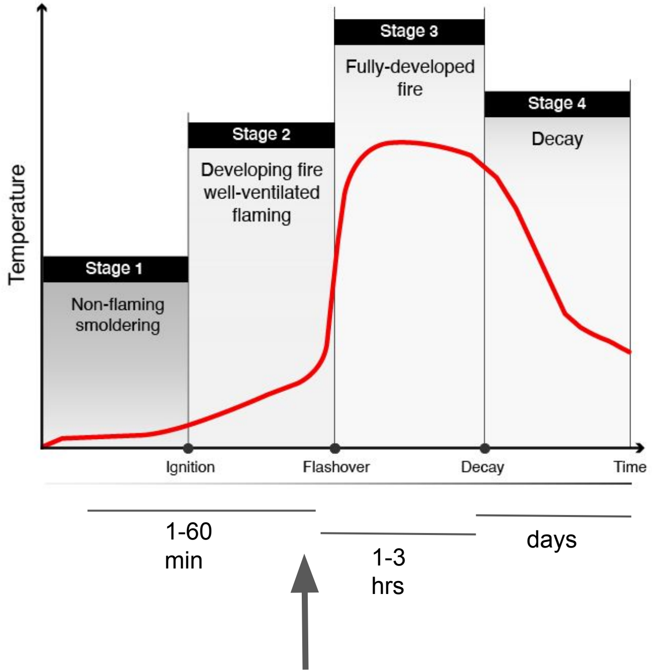
Use Case

- ❖ In 2022, California has had a total of ~ 6000 wildfires recorded, spanning 300K acres
- ❖ Notify wildland firefighters of locations of wildfires
  - ➢ Target audience: Fire & Emergency Services
- ❖ Project is designed to be scaled to cover an entire forest

ECE Area(s) covered: Software and Hardware Systems

Quantitative Use Case Requirements

- Fire Detection Accuracy
  - Goal: > 90-95 % accuracy
  - Why? We want our network to be able to detect fires accurately but also give slack in case one of the nodes in our network dies
- Notification Timing
  - Web application should show active fires within
    - within 30 mins of conflagration
- Low Power System
  - Nodes operate without maintenance for up to a month

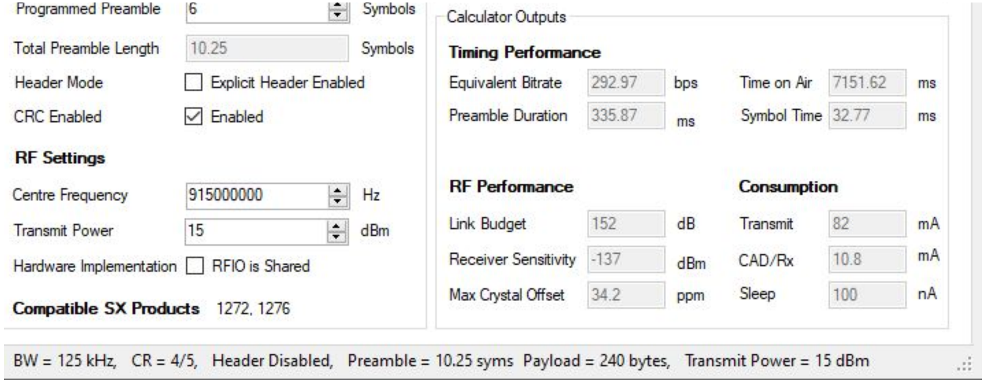


Address fire before it becomes an open fire

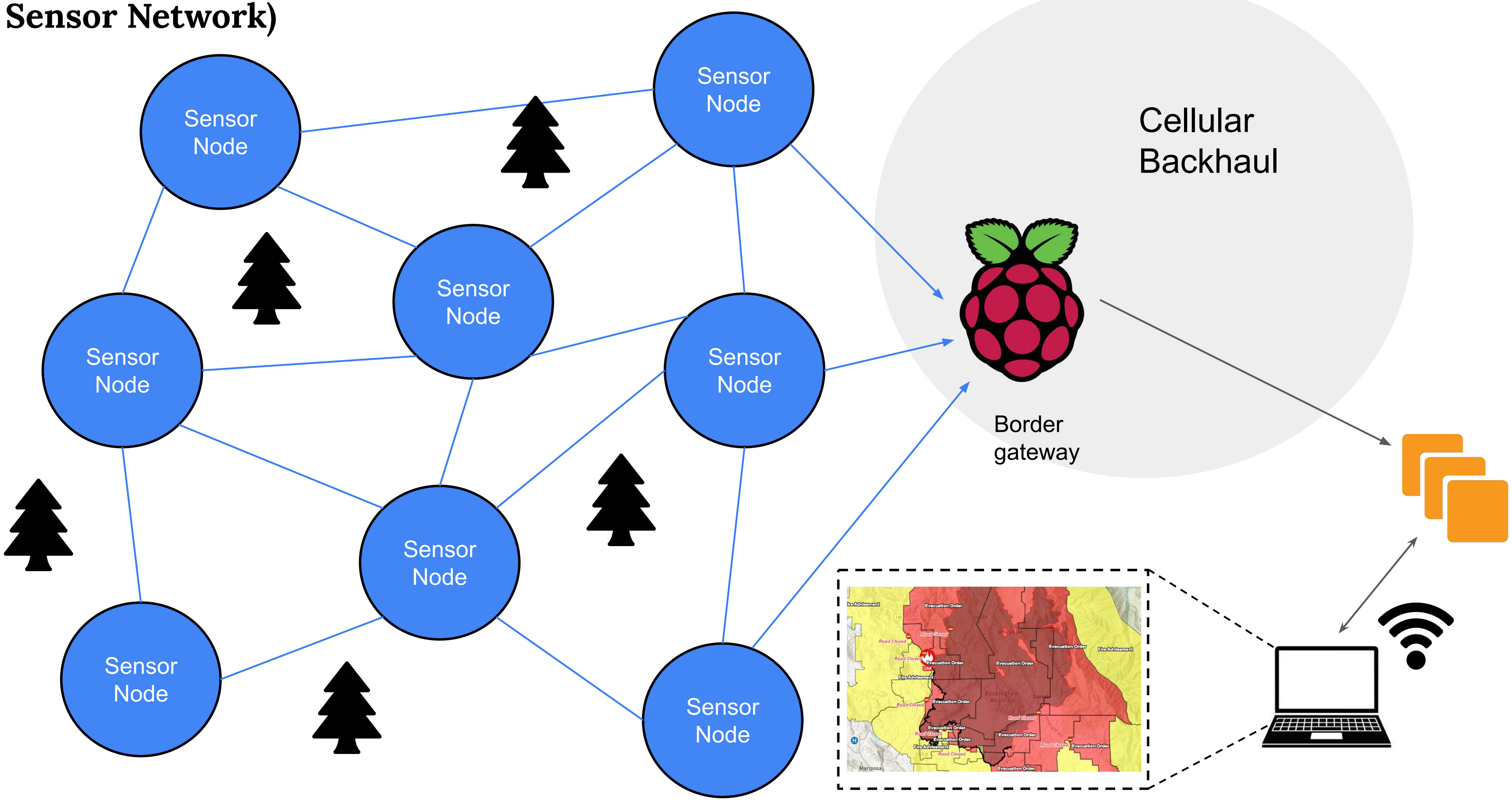
Source:
https://www.firetrace.com/fire-protection-blog/different-stages-of-a-fire

Solution Approach

- Wireless Sensor Network:
  - Measuring heat (and/or temperature) to determine if fire is occurring
    - Using 8 nodes to simulate our system
- Node Architecture:
  - STM32L4 for low power
  - LoRa Transceiver
  - Gas/temp sensor
- Routing/Network Protocol:
  - Time Division MAC protocol
    - Maximize low-power
- Web Application:
  - Hosting WebApp on RPi using Django software
    - Easy to use/well-documented software



System Specification
(Overall Sensor Network)

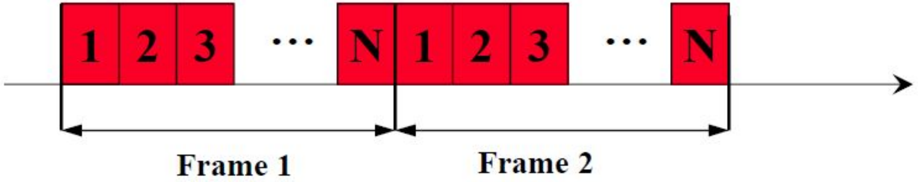


Source: https://www.mercurynews.com/wp-content/uploads/2022/07/oakfire0727.jpg?w=1024

MAC Protocol

- Goals: collision avoidance, energy efficiency, scalability and resilience
- Idle state power consumption (Nucleo board only):
  - 0.95 μA[no RX]
  - 158 μA [RX]
- Asynchronous B-MAC requires low-power listening
- Synchronous MAC requires clock synchronization
  - Gateway sends time-synchronization packets
  - Node-specific GPS for time synchronization

- LoRaWAN: star-of-stars topology, ALOHA
- Spanning Tree Protocol (STP): mesh network



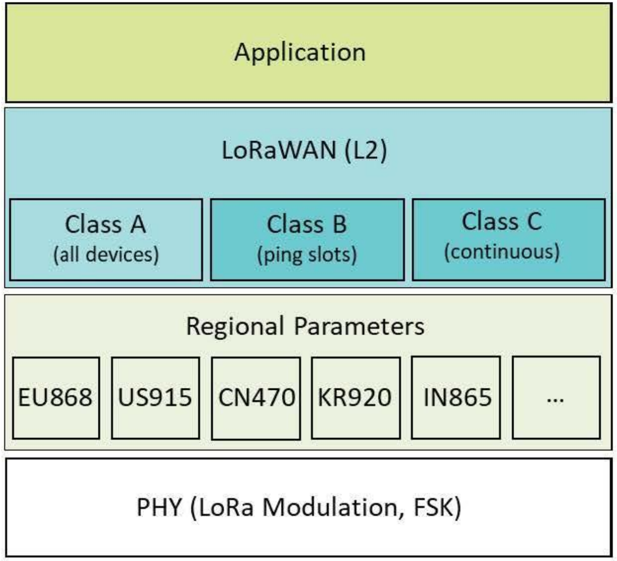


Figure 1: LoRaWAN Classes

Sources: Lecture 20, 18452 (Spring '21), LoRaWAN® L2 1.0.4 Specification

Routing and WSN simulation

Phase I. Mesh Network reduction

- mesh topology:
  - In a single loop-free path (GW to end-node), there cannot be more than two monotonously-decreasing nodes in between two monotonously-increasing nodes.
- STP runs on each node, tree topology converges in 3 iterations (rerun every 30 minutes)

Phase II. Data Transmission Scheduling, Clock Sync

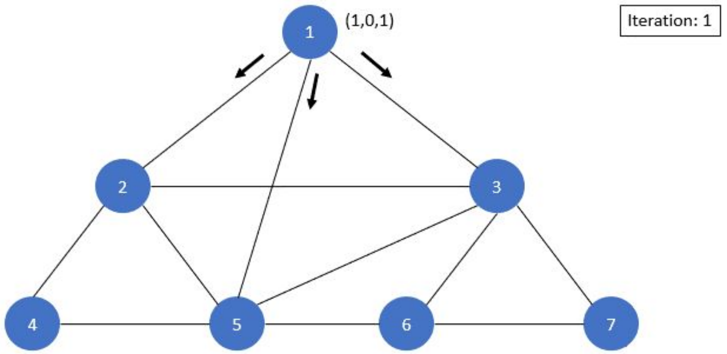
- 2N timeslots from the GW are subdivided and assigned by the parent node to its children nodes (if any exist)
- GW's timestamp is piggybacked onto the scheduling messages

Phase III. Sensor Data Collection

- Nodes follow their own data transmission schedules
  - unused timeslots recycled by leaf nodes

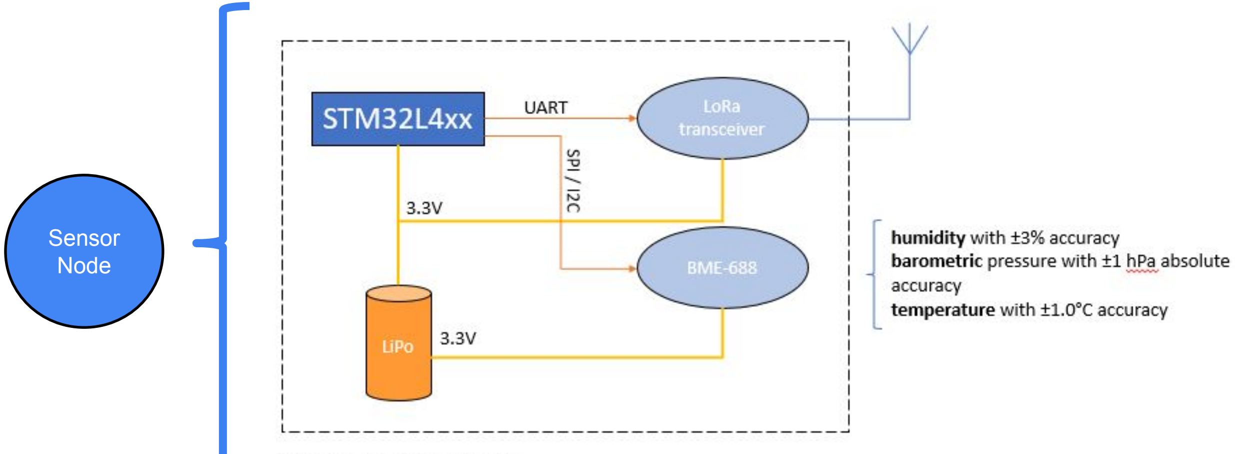
Routing and WSN simulation (cont'd)

Phase I. Mesh Network Tree reduction
Phase II. Data TX Scheduling, Clock Sync
Phase III. Sensor Data Collection



System Specification

(Node Architecture)



Node Architecture

Transceiver power consumption:

$[3(Q_{TX}+(Q_{RX})(N-2))] + [Q_{TX}+Q_{RX}] + [Q_{TX}+(N-2)(Q_{RX})] = 4.8759\ C$

Total duration: $[3N(TS)] + [(N+1)TS] + [(2N-3)TS] = 243$ seconds

2038.27mAh required for sampling rate of 1/30min

2588mAh required for sampling rate of 1/15min

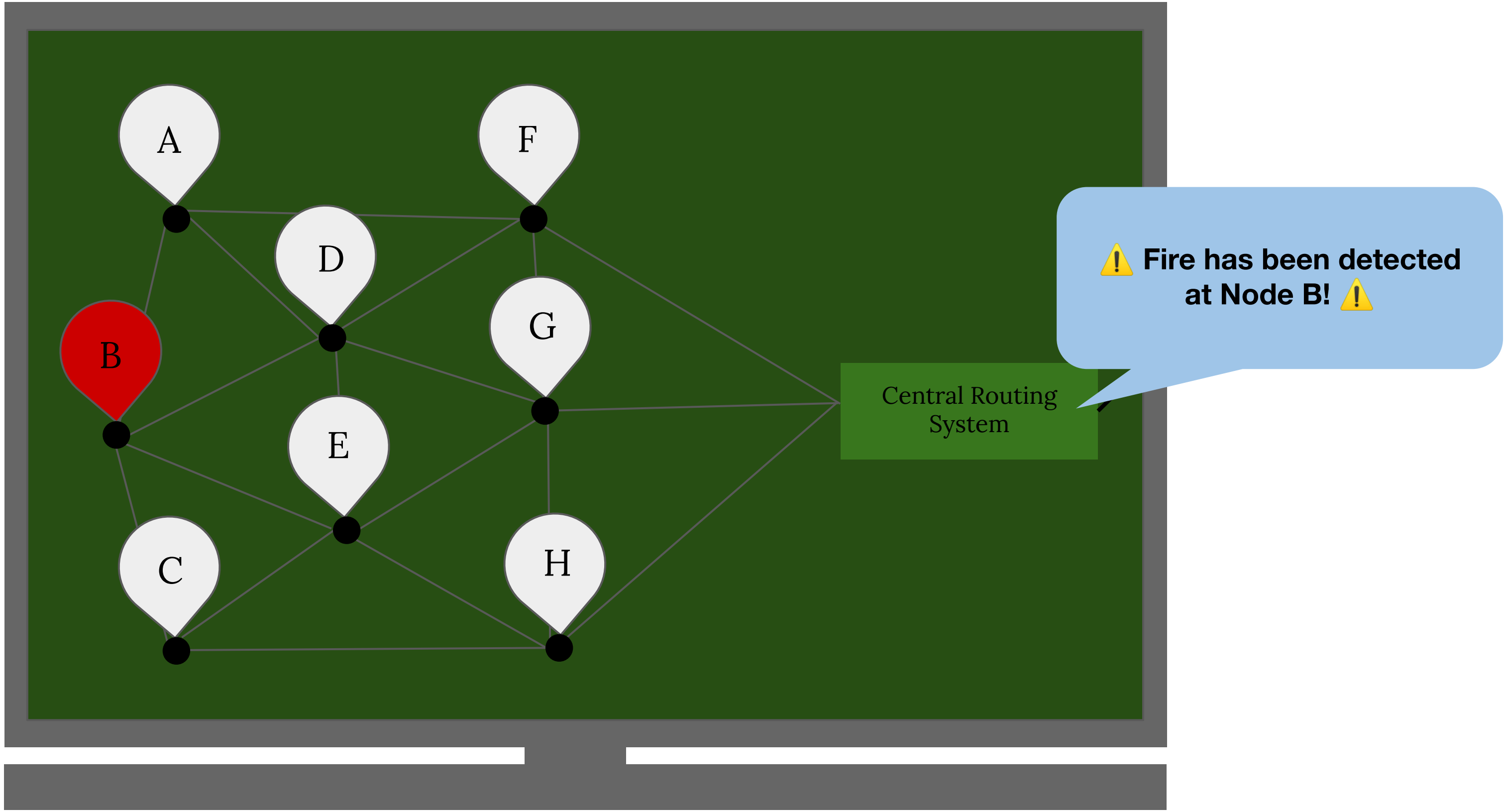
timeslot $TS = TOA + m_1$

$Q_{TX} = (I_{A\_TX})(t_s) = 586432.5 \times 10^{-6}\ C$

$Q_{RX} = 77751.4 \times 10^{-6}\ C$

Battery B = 3.3V @ 3000mAh

Web Application Design



Implementation Plan

- Routing/Networking Protocol
  - Adapting a time-division MAC protocol based off a clock-synchronization algorithm from *TDM MAC protocol design and implementation for wireless mesh networks* by Koutsonikolas and Salonidi
- Node Architecture
  - Designing the node's architecture from scratch
- Web Application
  - Designing from scratch using Django software
  - Will be using online resources as tutorials/reference

Test, Verification, & Validation

- Light a match next to a sensor, response time on the web application within 30 minutes of conflagration.
- Detection of (implicit) fire 30 minutes after node failure
- Maximize low power modes in the framework of our network protocol:
  - Turn nodes on for a period of time and measure how much battery is drained
  - Successful test: battery would last a month
- Successful test: fire is detected and shown on web application within 30 mins
- If a test fails:
  - If clock synchronization fails, we will use a GPS module to calibrate the RTC

Project Management

Tasks	Week 6	Week 7	Fall Break	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
	10/2	10/10	10/17	10/24	10/31	11/7	11/14	11/21	11/28	12/4
Class Assignments			-							
Project Proposal Presentation			-							
Design Presentation	DUE		-							
Design Document		DUE	-							
Ethics Assignment			-	DUE						
Interim Demo			-			DUE				
Final Preentation			-							DUE
Overall Project Work			-							
Individual Project Parts			-							
Connecting node architecture to router system			-							
Connecting router system to web app			-							
SLACK										
Web Application System			-							
Create a dummy web server to test Django			-							
Create the nodes and router interface to show										
Notification System working			-							
Connection with Router System			-							
Nodes/ Embedded System			-							
Put node into low power modes on a schedule			-							
Nodes listen, take measurement, and transmit data			-							
Nodes accept beacon and syncs clocks										
Nodes are synched and listen/transmit/sleep on schedule			-							
Routing Network System			-							
STP Protocol (Phase I) and Phase II (Data Schedule TX)			-							
Phase III (Data Mode for all nodes)			-							
Integration of protocol phases and low-power modes			-							
Testing of protocol stack and debugging			-							

Division of Taks	
Karen	
Ankita	
Arden	