Meshtastic Update

Presented to the South Pasadena Amateur Radio Club

January 7, 2026

What is Meshtastic?

- Adrian Pinedo, KN6MHG, gave a good overview of Meshtastic at our October 2024 meeting.
- Software and firmware that uses low-cost, low power radios for nodes to form a “mesh”
- Messages get broadcast by a node, and then rebroadcast by other nodes
- Allows propagation far beyond the line-of-sight by utilizing this message passing

How does this look to the end-user?

- Typically, the devices do not have a built-in UI (some do, but it's not the most common use case)
- Your Meshtastic device connects to your phone (or tablet) via Bluetooth. The user-interface resides in an app on your phone
- Messages entered into your phone are sent to your device, and then the Meshtastic device broadcasts the message
- Other nodes on the net will rebroadcast that message, and hopefully, it will get to its intended destination.

Meshtastic “Update”

- The promise of Meshtastic
- Hardware
- The reality of Meshtastic
- Is it fixable????
- Updating firmware is important
- Choosing a proper role for your node
- Understanding the routing algorithm(s)….
- Possible alternatives

Meshtastic is a work in progress…..

- Some of the behavior that I'll describe is only available in very recent firmware upgrades (2.7+)
- Some of the behavior described tonight may change in future firmware upgrades
- You only have control over that subset of nodes that you own
  - If your network includes nodes controlled by others, their choice to upgrade or not may affect the performance of the mesh network.

Why is Meshtastic so fascinating????

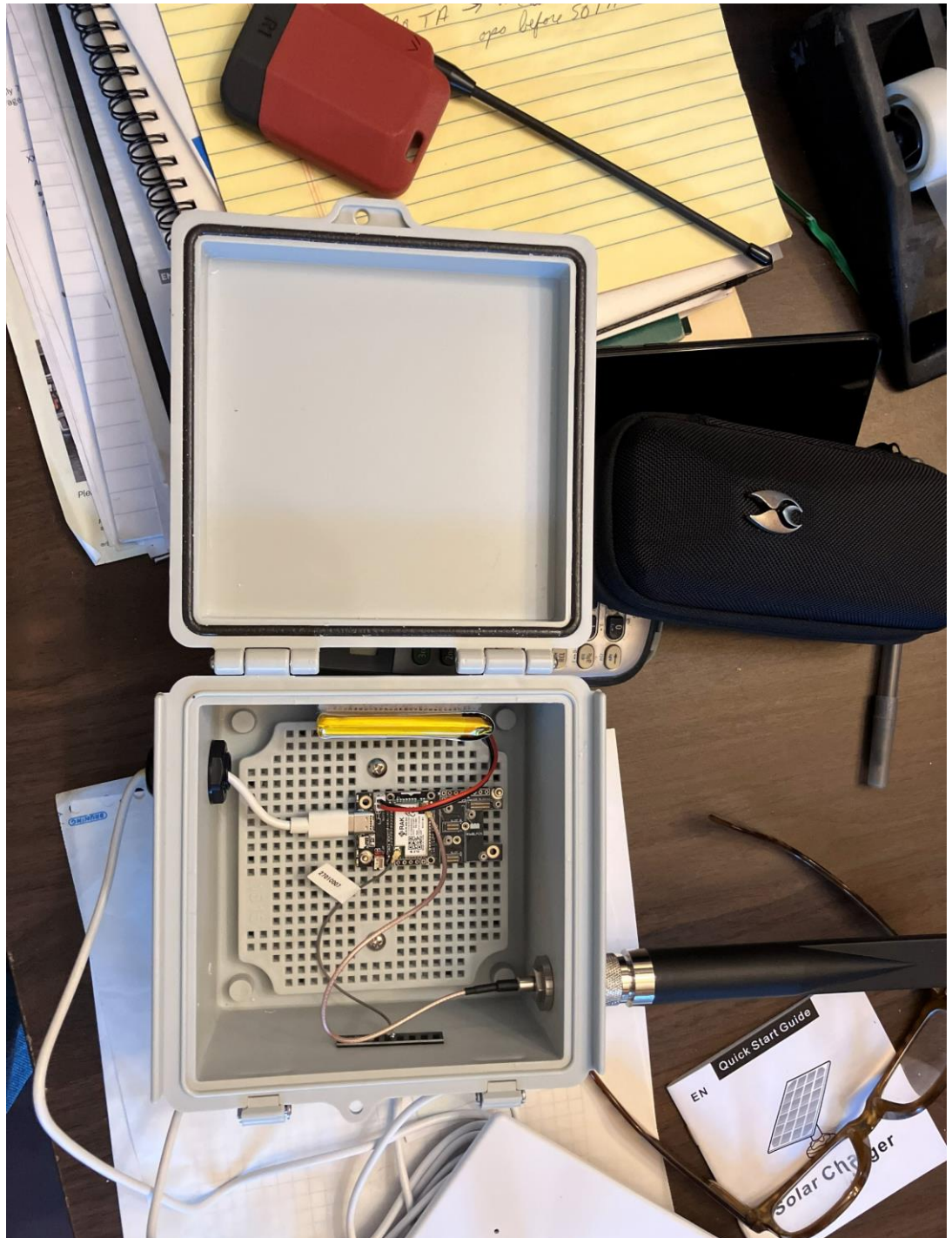
- Meshtastic offers the promise of low-power, regional (~50-100 miles), text-based communications
  - Independent of internet or phone networks
  - Potentially useful in emergency scenarios
- It does not require a license
- Uses inexpensive hardware

Low-cost hardware - boards

- Meshtastic boards typically range from $25 to $45
  - There are 3 main “families” which are distinguished by the microcontroller chip (MCU)
    - ESP32
      - Examples include the Heltec V3 and V4 boards, Lilygo T-Deck devices (which include a screen and keyboard – think of an old Blackberry device)
    - nrf52840
      - Examples include RAK 19007 board (the ones that we used for our build last year)
    - Raspberry Pi pico
  - All 3 families use the same radio chip

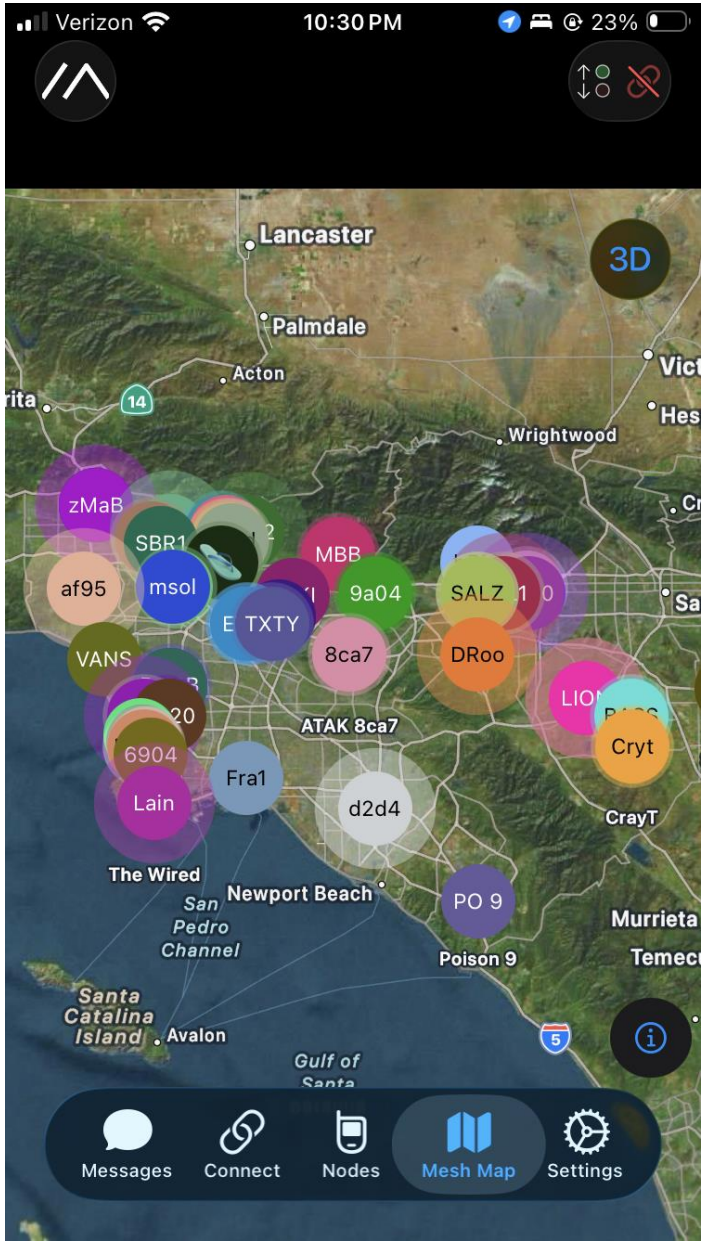
Low-cost hardware

- Handheld devices can be bought for $45-$100
- Weatherproof, solar nodes can be bought for $100-$250 (or built for <~$100)


Quick Start Guide
EN

A typical, new user experience?

- It’s easy to “see” lots of nodes on the default channel.



A typical, new user experience? (continued)

- Communication with those nodes that you see can be a bit of a random process
- Often not able to communicate with nodes that are relatively close. 1-3 miles.
- Can sometimes communicate with a specific node, but that same node is not reachable at other times

- **FRUSTRATION!**

What can be done to improve the situation?

- In order to improve performance, it will be necessary to dig into some details of how Meshtastic works
  - Understand the various “roles” that a node can have
  - Understand how a “Favorite” node can change things
  - Understand the routing algorithm
- Obviously, the presence of other nodes and the local topography also have an enormous impact on your ability to reach a specific node

What is a node “role”?

- A node that you carry around in your pocket should interact with other nodes differently than a node on the top of Mt. Wilson
- Meshtastic deals with these differences by defining different roles for nodes:
  - Client_mute
  - Client
  - Client_base
  - Router
  - Router_late
  - Sensor
  - Tracker
  - Plus other specialized roles…..

Node roles

- All roles except Client_mute may rebroadcast packets that they receive.
- Client is the most common role. It can be used for both fixed nodes and pocket nodes (personal communicator devices)
  - It may rebroadcast a packet if it does not hear another node rebroadcast
- Client_base nodes are often used in conjunction with client-mute nodes.
  - Example – Client_base node is on a house rooftop with the client-mute node used around the house and yard.
  - Client_base nodes make use of the “Favorites” designation.

Node roles (continued)

- Router role should only be used for fixed nodes in very favorable locations (i.e. locations that have a clear line of sight to a large area)
  - Always retransmit received packets.
  - Router_late is a special type of router that retransmits at the latest possible time after receiving a packet.
- Sensor and Tracker roles prioritize sending their telemetry or position data.
  - They can still forward other packets, but with a very low priority.

Meshtastic routing (high level description)

General "Managed Flooding" Logic

For most roles (such as `CLIENT`), nodes do **not** automatically rebroadcast every packet. Instead:

- **Listening Period:** When a node receives a packet, it waits for a short period before rebroadcasting.
- **Suppression:** During this wait, if the node hears another node already rebroadcasting that same packet, it will **cancel** its own scheduled rebroadcast to save airtime.
- **Prioritization:** Nodes with a weaker signal (lower SNR), typically those further away, wait for a shorter time. This encourages "long hops" to reach farther distances faster.

Meshtastic routing algorithm (routers)

Exception: ROUTER and REPEATER Roles

If a device is specifically set to the `ROUTER` or `REPEATER` role, its behavior changes to ensure high reliability for infrastructure:

- **Mandatory Rebroadcast:** These roles typically **always rebroadcast** every eligible packet they receive (unless the hop limit is already zero).
- **Preemption:** They use a very short contention window, meaning they "cut in line" to rebroadcast before other `CLIENT` nodes have a chance. This often causes nearby `CLIENT` nodes to hear the router first and suppress their own transmissions.
- **ROUTER_LATE:** This special role behaves like a `CLIENT` initially but will perform a "late" rebroadcast even if it heard others, ensuring a packet reaches local clusters that might have been missed.

Timing for packet rebroadcast

Summary of Rebroadcast Rules

Node Role	Rebroadcasts if others do?	Priority Level
CLIENT	**No** (Suppresses if heard)	Standard (SNR-based)
ROUTER / REPEATER	**Yes** (Always rebroadcasts)	High (Preempts others)
ROUTER_LATE	**Yes** (Rebroadcasts late)	Low (Polite/Gives way)
CLIENT_MUTE	**No** (Never rebroadcasts)	N/A

For more detailed technical specifications, refer to the Meshtastic Mesh Broadcast Algorithm documentation.

Packet rebroadcast timing….

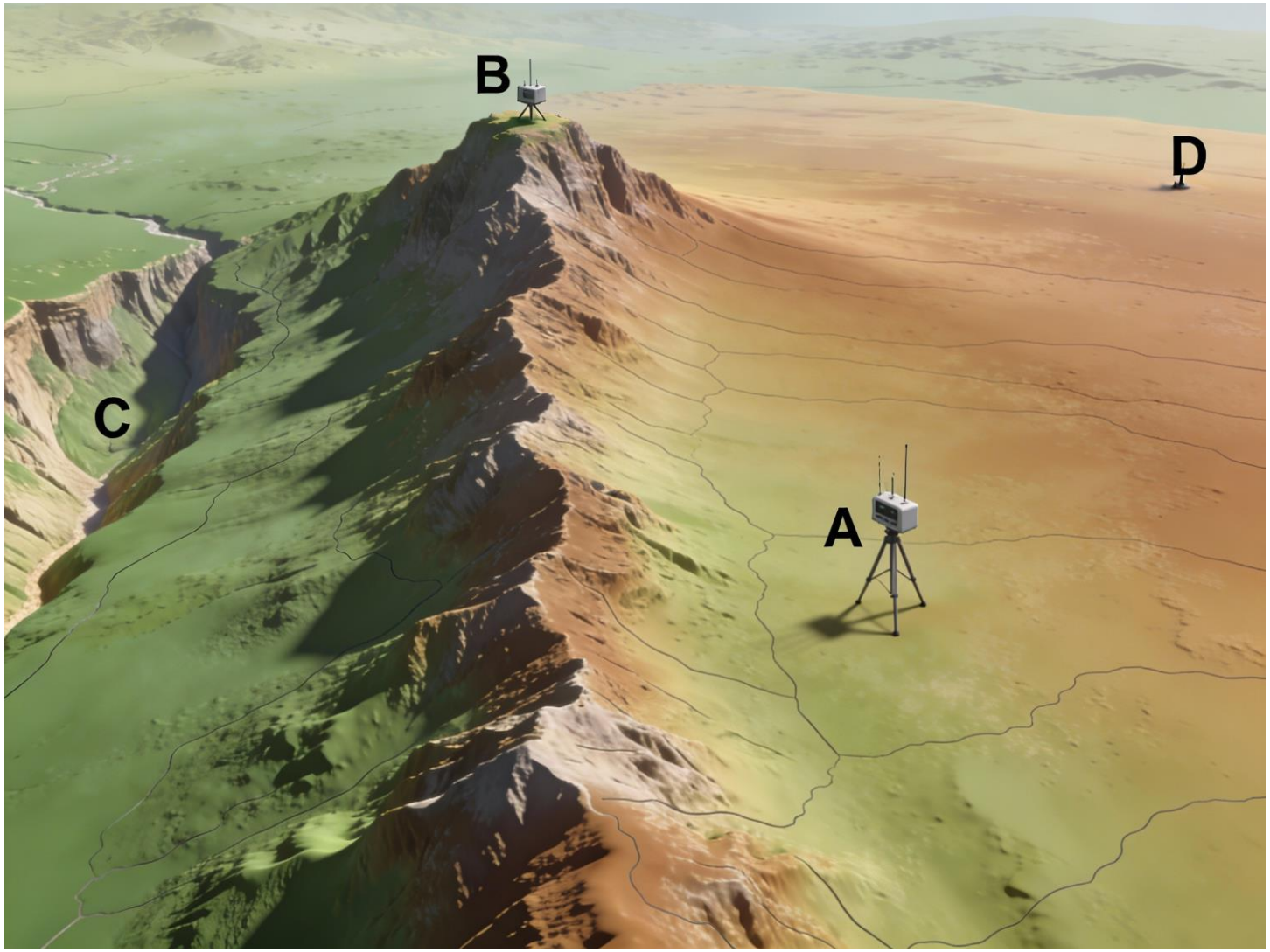
Rebroadcast Timing Mechanisms

- **Contention Windows**: Meshtastic uses non-overlapping time slices to determine when a node can transmit.
  - **Early Window**: Reserved for infrastructure roles like `ROUTER` and `REPEATER`. These nodes "cut in line" to ensure priority routing.
  - **Standard Window**: Used by standard `CLIENT` nodes. These nodes wait for a random multiple of "slot times" before attempting to rebroadcast.
- **SNR-Based Delay**: Within their contention window, a client node's exact delay is modified by the **Signal-to-Noise Ratio (SNR)** of the received packet.
  - **Low SNR (Distant nodes)**: Receive a **shorter** delay. This gives "first dibs" to nodes further away, allowing them to extend the message's reach more effectively.
  - **High SNR (Close nodes)**: Receive a **longer** delay. This prevents nearby nodes from saturating the local airtime.
- **Managed Flooding Cancellation**: If a client node hears another node rebroadcast the same packet while it is still waiting in its contention window, it will **cancel** its own rebroadcast to save airtime.

Why is message transmission so unpredictable?

- Most Meshtastic users stay on the default Long-Fast setting.
  - This gives their node access to more potential nodes to connect to
  - However, the mesh repeatability is affected by….
    - The very small number of routers.
      - We are strongly discouraged from setting up a router node, because having too many routers will degrade the network performance by flooding the frequency with traffic
    - The nature of the SNR prioritization used for client mode rebroadcast has a bias toward distance, but may cause nearby nodes to be missed….

Imagine this case…



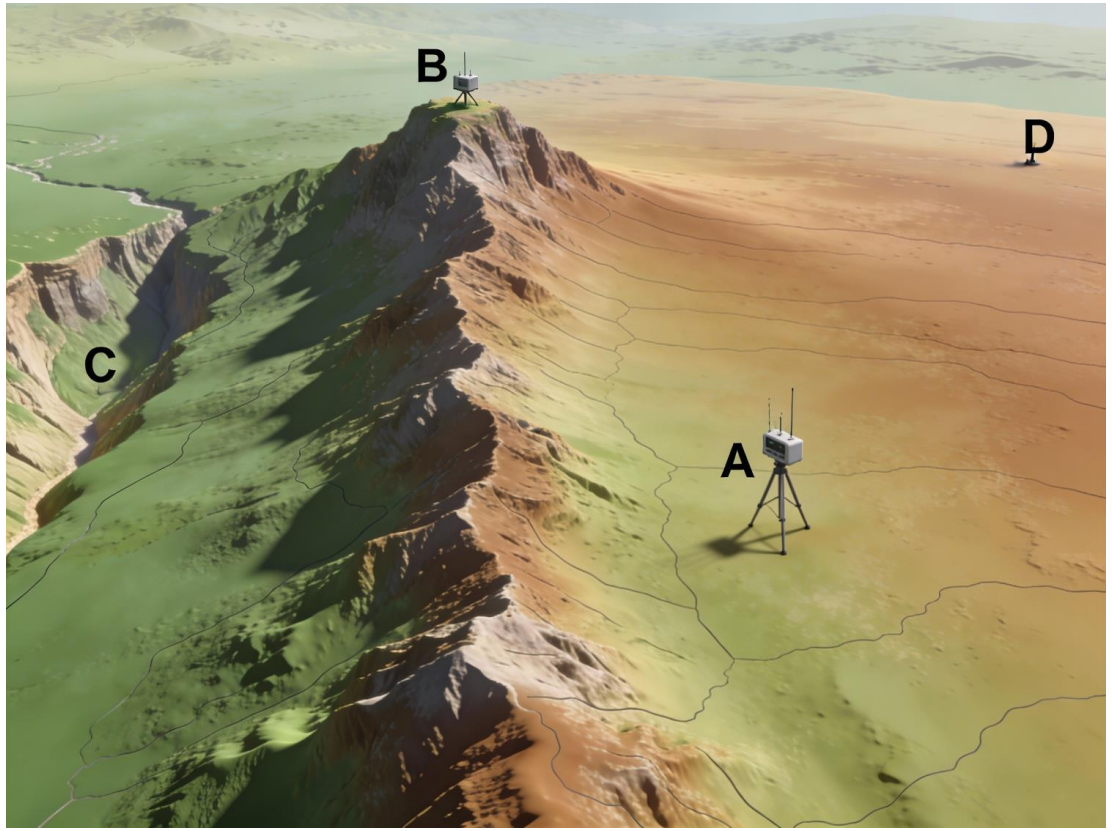
Favorite nodes and the Client_Base role

- Nodes can be designated as “favorites.”
- For most roles, the only effect of designating another role as favorite is that it won’t be deleted from the node list.

- For Client_Base nodes, packets from favorite nodes are rebroadcast in the “early window” -- effectively turns the Client_Base node into a router for its favorite nodes.
  - This can make routing much more predictable if it’s possible to configure a set of Client_base nodes along a route from one node to another as favorites of each other

How does our failed case look with Client_Base and favorites?

- Node A originates the message
- Node B, which is designated as a Client_Base node with Nodes A and C in its favorites list will quickly rebroadcast the message before Node D (still just a Client) has a chance to do so.
- Node C will hear the rebroadcast message
- **Success!**



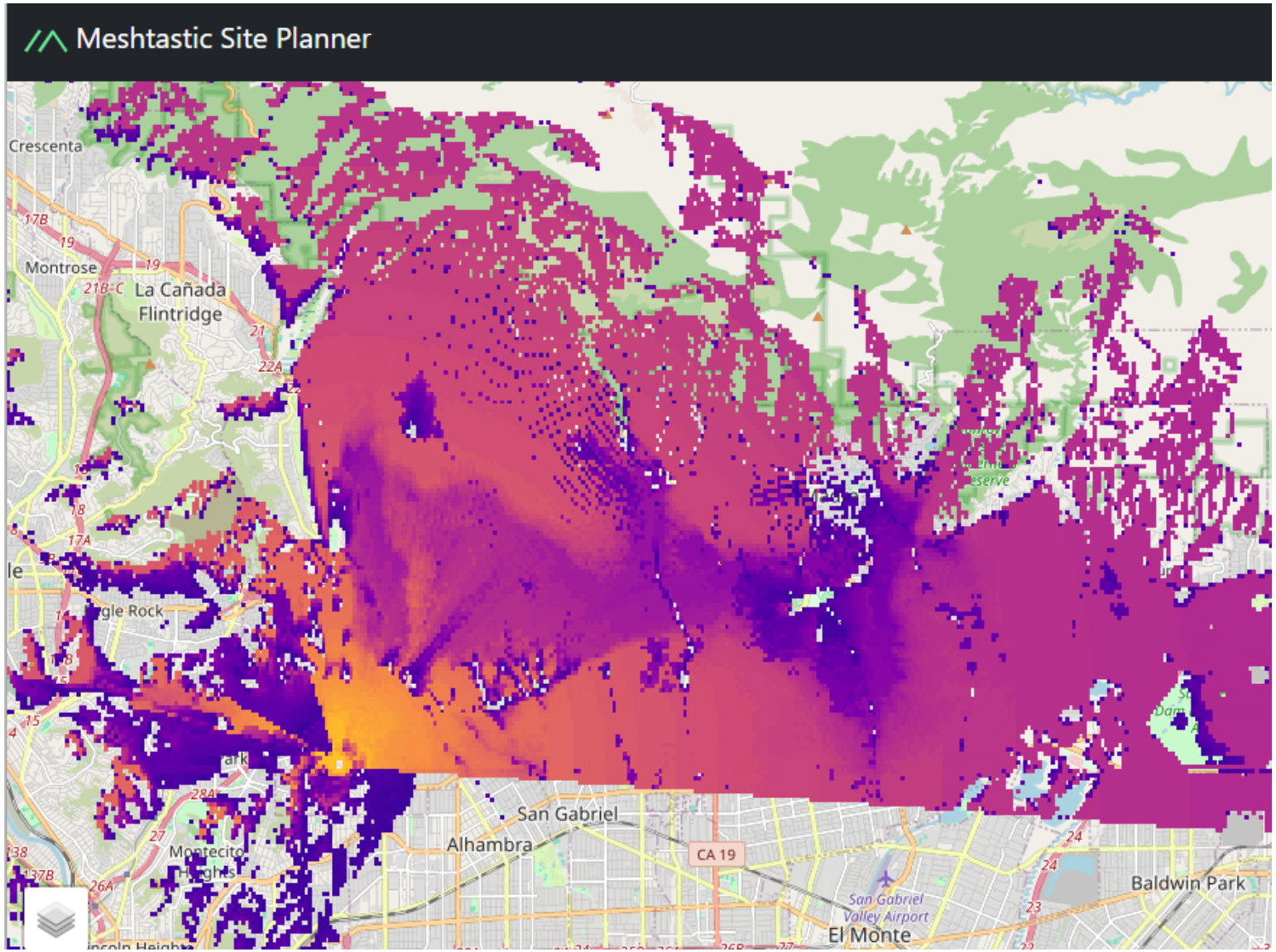
Can we use this feature now?

- It is available on Firmware versions 2.7 and beyond
- **Update your firmware**

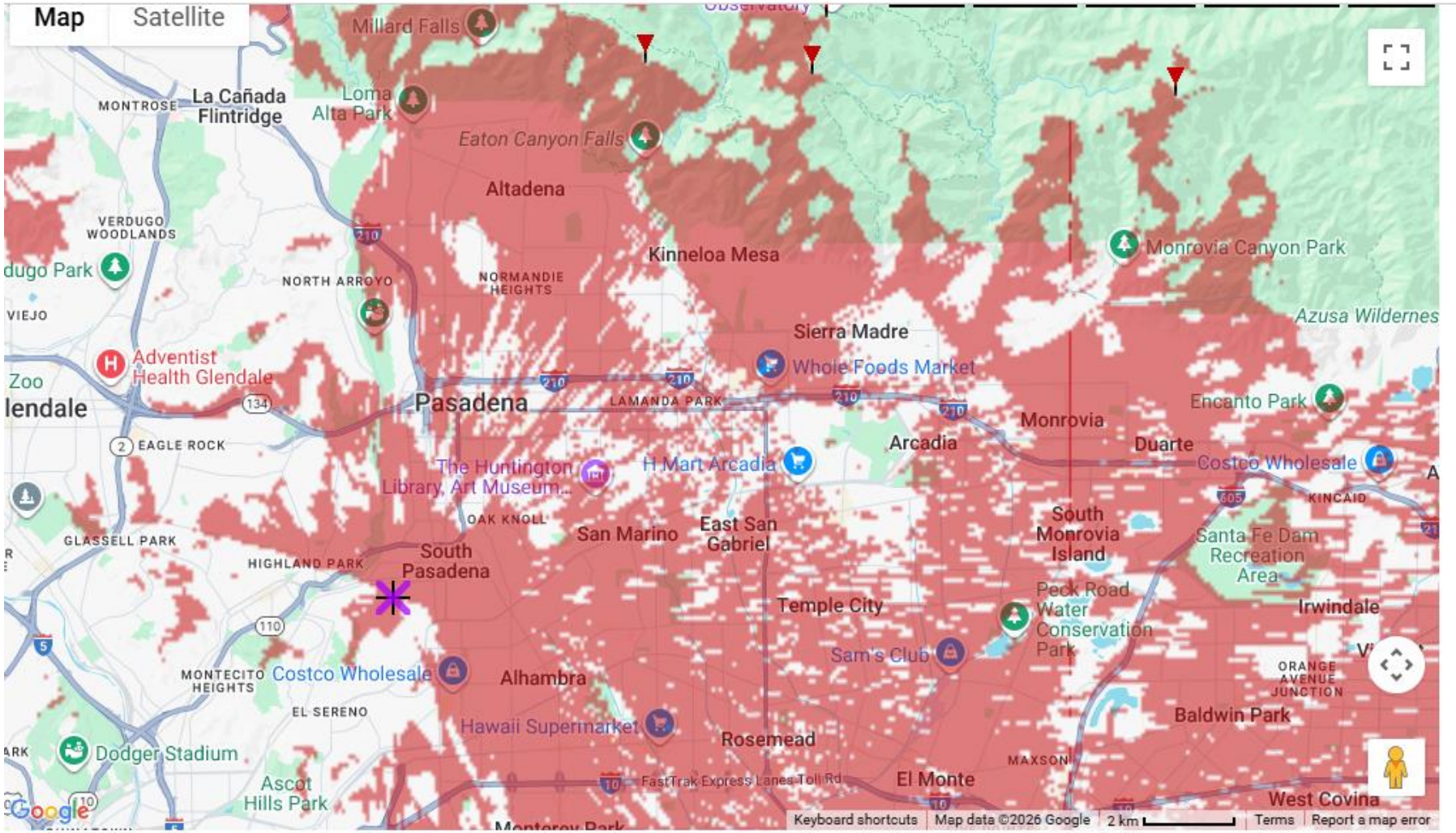
Location, location, location

- After all that has been said, the strategic placement of nodes is still the most important step to getting a robust network.
  - Obviously, we are limited to locations which we own/control, or for which we have been granted permission to use.
- Using mapping tools can help us to determine the best locations:
  - Heywhatsthat.com
  - Site.meshtastic.org

Site.meshtastic.org



HeyWhatsThat



34.1088 N 118.163 W 894 ft

Click here to re-center map on South Pasadena Water Tower

English Metric DD.DDDDDD° DD° MM.MMMM' DD° MM' SS.SS"

decimal places (0-6) 5 Pan to South Pasadena Water Tower or find

Some other observations on Meshtastic architecture

- Many of the nodes are broadcasting telemetry data (position, battery state of charge, channel utilization…)
  - This telemetry takes up a fair chunk of time, limiting the time available for text messaging.
  - A recent look at the channel utilization for the Long-Fast (default) frequency band (Saturday night) showed it to be at 40%
    - In all likelihood, in the event of a disaster, this frequency band would be totally useless due to excessive demand and contention.
    - So…it probably makes to set up local nets using a frequency slot other than slot 20 (the default slot for Long-Fast setting) --- BUT, this isolates your network from the vast majority of nodes. You'll have to make sure that you have enough nodes to make an effective mesh.

For “out of the box” usage (i.e. no tweaking of default settings)

- It is best for small groups who are off grid, but in close proximity
  - A small group of hikers who may have short separations between them can keep in touch
  - A family at a large, outdoor event (for example, a stadium, a park etc. -- Coachella!)

Can Meshtastic be made reliable???

- For reliable emergency communications, the jury is still out, but there are some ways to fine tune the node settings and locations to improve the default (very unreliable) behavior

Meshcore is an alternative which may be more suitable for some scenarios

- It runs on many of the same hardware devices as Meshtastic.
- Roles are Repeater, Companion and Room Server
  - Repeater – retransmits messages.
  - Companion – Similar to Meshtastic's Client_Mute. Can send and receive messages, but does not rebroadcast packets
  - Room Server – Stores messages; Companion nodes that have been disconnected from the mesh can retrieve stored messages when they reconnect.
- Telemetry data is **NOT** sent periodically (though it can be requested)

Meshcore “philosophy”

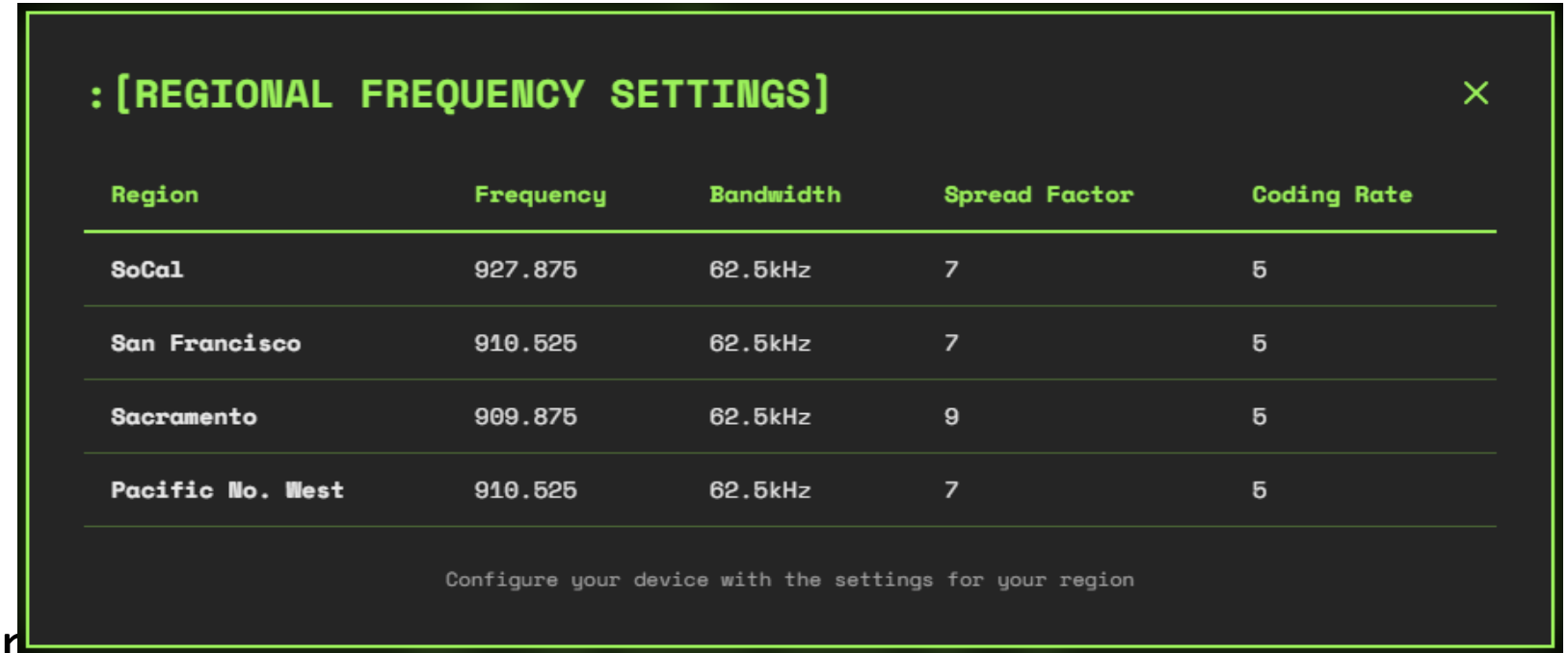
- Repeaters should be fixed infrastructure
  - Having mobile units serve as repeaters reduces predictability, reliability of the mesh.
- Passing messages between companion nodes is the most important job of the mesh.
  - Information on node health, location etc. is less important
- Users should not have to remain continually connected to the network to get messages.

Does it work???

- There are widespread deployments in Europe, and users report better reliability of message transmission.
- There are well-developed Meshcore meshes in the Pacific northwest and Austin that have gotten positive reviews from users.
- There is plenty of anecdotal evidence that it is more robust than Meshtastic (or at least, Meshtastic meshes as they are typically configured.)

Caveats

- Configuring a Meshcore node involves more effort than a Meshtastic node.
  - Just going with the presets will not work in SoCal –



:[REGIONAL FREQUENCY SETTINGS]

Region	Frequency	Bandwidth	Spread Factor	Coding Rate
SoCal	927.875	62.5kHz	7	5
San Francisco	910.525	62.5kHz	7	5
Sacramento	909.875	62.5kHz	9	5
Pacific No. West	910.525	62.5kHz	7	5

Configure your device with the settings for your region

- In Southern California, the Meshcore network is less developed, but that is changing rapidly.

Conclusions

- Many users have been frustrated by Meshtastic but…
  - It’s too early to give up on the technology
  - Firmware updates are enabling more intelligent deployment configurations
  - The knowledge base in the community is growing

- If Meshtastic continues to prove to be unreliable, Meshcore seems to offer some hope of a similar capability that may prove more reliable.

Questions