

A Rover-Friendly APRS Propagation Tool

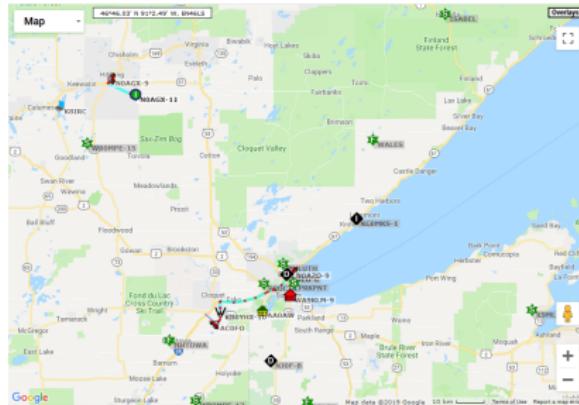
Bill Mitchell, AEØEE

Aurora VHF+ Conference

April 27, 2019

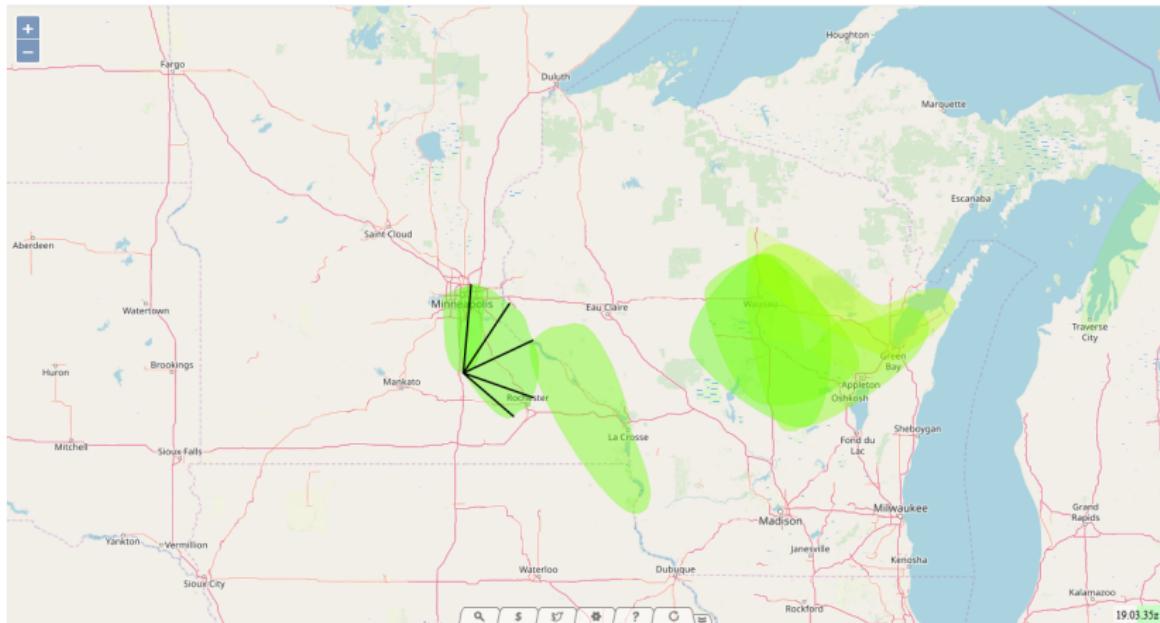
APRS Propagation Reports

- Unlike HF, little FT8 on 2m
- Automatic Packet Reporting System (APRS)
- APRS is digital, with position
- Automatic digipeater stations beacon frequently
- Use APRS (data on FM) as propagation tool.



Existing Tools

- NGØE: aprs.mountainlake.k12.mn.us



Challenges and Solutions

- Challenges

- Need internet access!
- APRS Internet Service filters out duplicates
- Throwing away good data

- Solutions

- Locally-based RX
- Runs on Raspberry Pi (RPi3)
- Web-based display



Backend Overview

- Decode ([direwolf](#))
- Parse ([aprs-python](#), custom)
- Database
(Postgres/PostGIS)
- Workflow ([aprsdb](#))

```
def process_digi(parsed, cur):
    """Insert values for a new digipeater from a parsed packet
    parsed: an aprslib parsed packet
    cur: psycopg2 database cursor
    """
    # Check for crucial information
    if ('symbol' not in parsed.keys()):
        raise KeyError("Missing key: symbol")

    # Get existing record for digi
    cur.execute("SELECT digi_id, name_sym FROM digits WHERE call=%s;", (parsed['call'],))
    if (cur.rowCount == 0): # No matching digi
        insert_digi(parsed, cur)
    else: # Digits are forced unique by callsign, so can only be one row matching
        myresult = cur.fetchone()
        # Check if updates are needed and make them
        check_update_digi(parsed, myresult[0], cur)

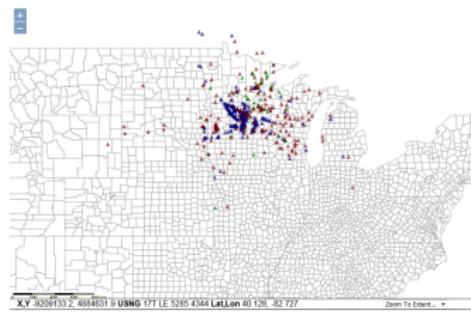
def remove_NULL_path(path):
    """
    Remove NULL entries from APRS path
    path: path of parsed packet (list of strings)
    returns: list of digipeaters with NULL entries removed, in order
    """
    digipath = []
    for digi in path:
        if digi!="NULL":
            digipath.append(digi)
    return(digipath)

def process_path(path, packet_id, conn):
    """
    Insert the path routing info to the database
    path: Python list of path routing elements (e.g. NOQVC-1, WIDE1*, NOBPA-1,WIDE1*)
    packet_id: packet_id from packets table corresponding to the entry being processed
    conn: psycopg2 database connection
    """
    cur = conn.cursor()
    path = aprslib.util.remove_WIDE_N(path) # Get rid of WIDE-N and asterisks
    if 'NULL' in path:
        path.remove('NULL') # remove first NULL value

    digi_src = False
    cur.execute("SELECT id,call FROM common AS t1 INNER JOIN digits AS t2 ON t1.id=t2.src WHERE t2.pid=%s;", (packet_id, )) # Check if source is a known digi
    all=cur.fetchall()
    221,5 33%
```

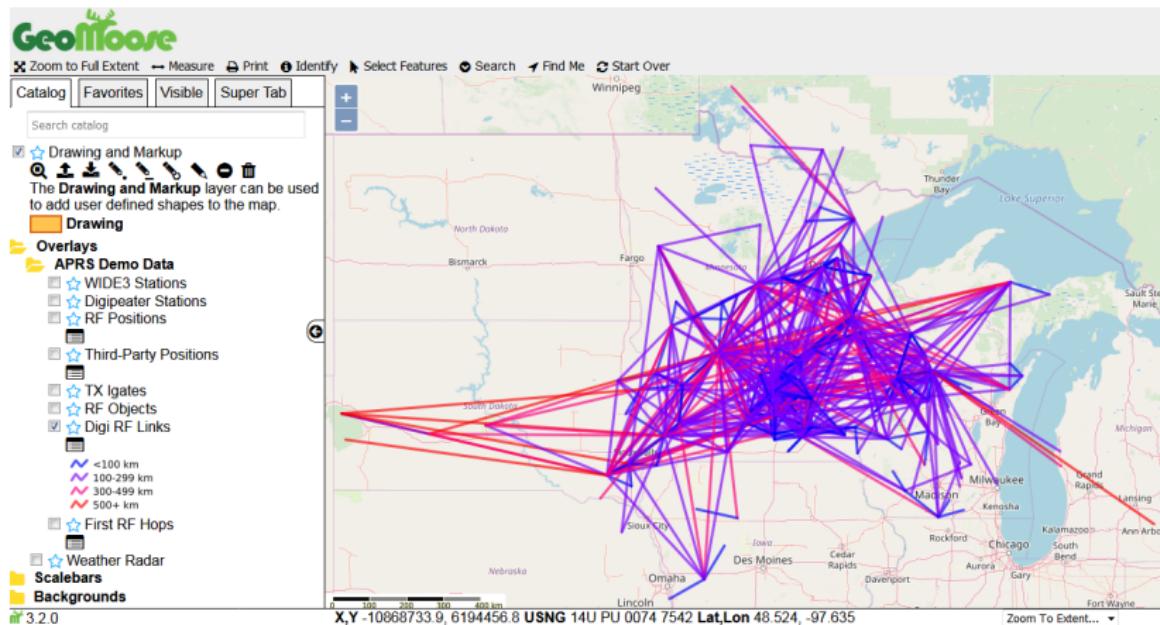
Frontend

- Interactive mapping ([QGIS](#))
- Local web server (Apache)
- Web front-end ([GeoMoose](#))
- Locally-stored county boundaries
- OpenStreetMaps possible
 - Network
 - Cached in advance
- Demo site ([Test data](#))



Demonstration

● Demonstration time!



Summary

- Pulls in and displays APRS data
- Usage statistics
- Runs on RPi3
- Web front-end
- All local—rover friendly
- Future work
 - Path location overhaul
 - Clean up web query code
 - Time-based queries
 - Data rotation (limit size!)



A Rover-Friendly APRS Propagation Tool

Bill Mitchell, AEØEE

Aurora VHF+ Conference

April 27, 2019