Realtime Multicast for SDR Interconnect

• Phil Karn, KA9Q
Some background

• Retired from Qualcomm Sept 2011

• Second career as volunteer mentor to San Diego high school and university ham clubs
  • ham licensing
  • high altitude balloons
  • cubesats
KA9Q-radio

• Set of general purpose SDR modules
  • multicast proof of concept
• Minimum cost (I work with students)
  • RPIs, Funcube dongles, HackRF, etc
    • must be open, cheap and available
• Balloon APRS, satellite operations
Is it real time?

• “Real Time” != “audio and video” !!

• Real time is **real time** — *is the stream being generated right now? Is latency important?*

  • Use RTP for real time data, too: AX.25, etc.

  • Just use TCP for recorded video/audio
RTP

- Stable Internet standard for real time streams
- multicast or unicast
- VoIP, IPTV
- Why not use it for SDR interconnection?
RTP features

- Runs above User Datagram Protocol (UDP)
- Sequence number
  - detect packet loss
- Timestamp - counts samples, frames, etc
  - can be discontinuous
- Payload type, stream source ID, mark flag
KA9Q-radio modules

- funcube
- hackRF
- radio
- opus
- monitor
- packet
- aprs
- aprsfeed
- iqplay/iqrecord
KA9Q-radio for UCSD balloon flights

- funcube
  - iq.vhf.mcast.local
    - iqrecord
      - pcm.vhf.mcast.local
        - opus
          - opus.hf.mcast.local
            - monitor
              - ax25.mcast.local
                - aprs
                  - antenna rotors
                - aprsfeed
                  - APRS net
## Radio Screenshot - VHF (HackRF)

<table>
<thead>
<tr>
<th>Tuning</th>
<th>Signal</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier</td>
<td>IF</td>
<td>Receiver profile: FM</td>
</tr>
<tr>
<td>147,435.000.000 Hz</td>
<td>-111.4 dB</td>
<td>Band: 2m</td>
</tr>
<tr>
<td>Center</td>
<td>Baseband</td>
<td>Emissions: Voice Image Data CW</td>
</tr>
<tr>
<td>147,435.000.000 Hz</td>
<td>-111.5 dB</td>
<td>Privs: Extra Adv Gen Tech</td>
</tr>
<tr>
<td>First LO</td>
<td>N0</td>
<td></td>
</tr>
<tr>
<td>147,483,000,000 Hz</td>
<td>-181.5 dB/Hz</td>
<td></td>
</tr>
<tr>
<td>IF</td>
<td>S/N0</td>
<td></td>
</tr>
<tr>
<td>-48,000,000.000 Hz</td>
<td>70.0 dB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NBW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>42.0 dBHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SNR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>27.9 dB</td>
<td></td>
</tr>
</tbody>
</table>

### Filtering

<table>
<thead>
<tr>
<th>Low</th>
<th>High</th>
<th>FM demodulator</th>
<th>Options</th>
<th>SDR Hardware</th>
<th>Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-8,000,000 Hz</td>
<td>+8,000,000 Hz</td>
<td>Loop SNR</td>
<td>ISB</td>
<td>Samprate</td>
<td>FM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Offset</td>
<td>PLL</td>
<td></td>
<td>FMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Deviation</td>
<td>Square</td>
<td></td>
<td>AM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tone</td>
<td>Mono</td>
<td></td>
<td>CAM</td>
</tr>
<tr>
<td>Blocksize</td>
<td></td>
<td></td>
<td>Stereo</td>
<td></td>
<td>DSB</td>
</tr>
<tr>
<td>3,840</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IQ</td>
</tr>
<tr>
<td>FIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ISB</td>
</tr>
<tr>
<td>4,353</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CISB</td>
</tr>
<tr>
<td>Freq bin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CWU</td>
</tr>
<tr>
<td>23.438 Hz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CWL</td>
</tr>
<tr>
<td>Delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>USB</td>
</tr>
<tr>
<td>31.333 ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LSB</td>
</tr>
<tr>
<td>Interpolate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AME</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### I/O

- Source: 192.168.42.4:54957 -> iq.hackrf.mcast.ka9q.net SSRC 5b97a610
- IQ pkts 153,496,504 samples 53,723,776,750 drops 1
- Time: Fri Sep 14 17:08:44.521391 UTC 2018
- Sink: pcm.hackrf.mcast.ka9q.net; ssrc 5b978fe6; TTL 1
- PCM 48,000 Hz; pkts 13,587,643

KA9Q SDR Receiver v1.0; Copyright 2017-2018 Phil Karn
Compiled on Sep 11 2018 at 02:48:35
Radio screenshot - HF (WWV)

Carrier 10,000,000.000 Hz
Center 10,000,000.000 Hz
First LO 9,951,999.850 Hz
IF 48,000.150 Hz

IF -82.7 dB
Baseband -85.5 dB
N0 -136.5 dB/Hz
S/N0 58.8 dBHz
NBW 37.8 dBHz
SNR 13.0 dB

Receiver profile: cam
Band: WWV 10 MHz

Low -3,000.000 Hz
High +3,000.000 Hz
Shift +0.000 Hz
Beta 3.000
Blocksize 3,840
FIR 4,353
Freq bin 23.438 Hz
Delay 31.333 ms
Interpolate 1
Decimate 4

Source: 192.168.42.67:59971 -> iq.hf.mcast.local SSRC 5b977b87
IQ pkts 228,559,539 samples 54,854,290,320 drops 222 dupes 218
Time: Fri Sep 14 17:11:51.212577 UTC 2018
Sink: pcm.hf.mcast.local; ssrc 5b978fc8; TTL 1
PCM 48,000 Hz; pkts 28,569,890

KA9Q SDR Receiver v1.0; Copyright 2017-2018 Phil Karn
Compiled on Sep 11 2018 at 02:48:35
Audio monitor screenshot

<table>
<thead>
<tr>
<th>Type</th>
<th>ch</th>
<th>BW</th>
<th>Gain</th>
<th>Pan</th>
<th>SSRC</th>
<th>Queue</th>
<th>Source/Dest</th>
<th>Opus packets</th>
<th>Source/Dest packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opus</td>
<td>20</td>
<td>ms</td>
<td>2</td>
<td>20</td>
<td>+0 +0.0</td>
<td>5b978fc8</td>
<td>0.03 homer.local:54638 -&gt; opus.hf.mcast.local</td>
<td>14,642</td>
<td></td>
</tr>
<tr>
<td>Opus</td>
<td>20</td>
<td>ms</td>
<td>2</td>
<td>20</td>
<td>+0 +0.0</td>
<td>5b978fe6</td>
<td>0.01 homer.local:37317 -&gt; opus.hackrf.mcast.local</td>
<td>11,077</td>
<td></td>
</tr>
<tr>
<td>Opus</td>
<td>20</td>
<td>ms</td>
<td>2</td>
<td>20</td>
<td>+0 +0.0</td>
<td>5b978fff</td>
<td>-0.20 homer.local:51622 -&gt; opus.vhf.mcast.local</td>
<td>5,445</td>
<td></td>
</tr>
</tbody>
</table>

M-b~GM-% select next stream
M-b~F-% delete stream
M-b~F-Q volume +1 dB
M-b~F-S volume -1 dB
M-b~F-R stereo position right
M-b~F-P stereo position left
Opus Codec

• Xiph + Skype merged algorithms
• IETF standard, many players
• 6 - 510 kb/s: comm voice to high fi
• Excellent reference implementation
• Free and open!
Wifi: a show stopper?

• Many consumer access points roll over and die when they see fast multicast streams, even with no WiFi clients listening

• *sigh*
Multicast & WiFi

- Ever-growing list of modulation and coding schemes (MCS) and MIMO (lots of antennas)
- unicast works great: dynamic MCS + acks
- multicast: slow and unacked
  - very poor performance - can kill an AP!
Fixing WiFi multicast

• IGMP snooping in switches
• Multicast-to-unicast conversion
  • AP sends acked unicast to each group member
• Radio is no longer a broadcast medium!
Observations

- Successful experiment: RTP works well
- Small modules with simple text UIs
- Multicast over small wired LANs works great
- Multicast over WiFi is a serious problem
- Wide area multicast is difficult
  - tunneling, routing often required
• Code is on https://github.com/ka9q/ka9q-radio
• All open source (of course)
• C, some Intel SIMD (eg. decimation)
• Runs on Linux (x86-64, RPi) & OSX
• Collaborators welcome!
Near-term ideas

- Turnkey APRS iGate (Rx only)
- Multicast inputs for WSJT, etc
- More digital demods: DMR, D*Star, Fusion
- BPSK satellite modems
- Medium speed UHF terrestrial modem
- Automatic satellite downlink recording
• Ham comm programs (WSJT, etc) can accept multicast streams
  • no need to use computer sound system!

• APRS needs a serious overhaul

• We can do much better than DMR/Fusion/D*Star

• IP multicast ideal for “tactical” (round table)

• We can build one!
Longer-term ideas

• We can do much better than DMR/D*/Fusion

• Proprietary codecs are evil!

• Inflexible network layers

• C4FM is inexcusably inefficient
  • 1200 Hz spacing @ 4800 baud??!?!?!?

• APRS badly needs an overhaul
Ham Multicast?

• Digital voice with CODEC2 + Opus
  • vary data rate with available capacity

• Round table operation
  • A multicast group is a “talkgroup”
  • Much better user interfaces are possible

• Easily support metadata: e.g., positioning