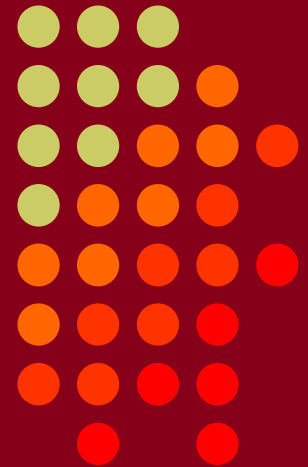


CTU 2023 Presents

How to Maximize Your Digital
Contest Station and Operation

Ed Muns, W0YK & P49X



• CTU •
CONTEST
UNIVERSITY

ICOM®

Maximizing Digital Contesting



● RTTY Contesting

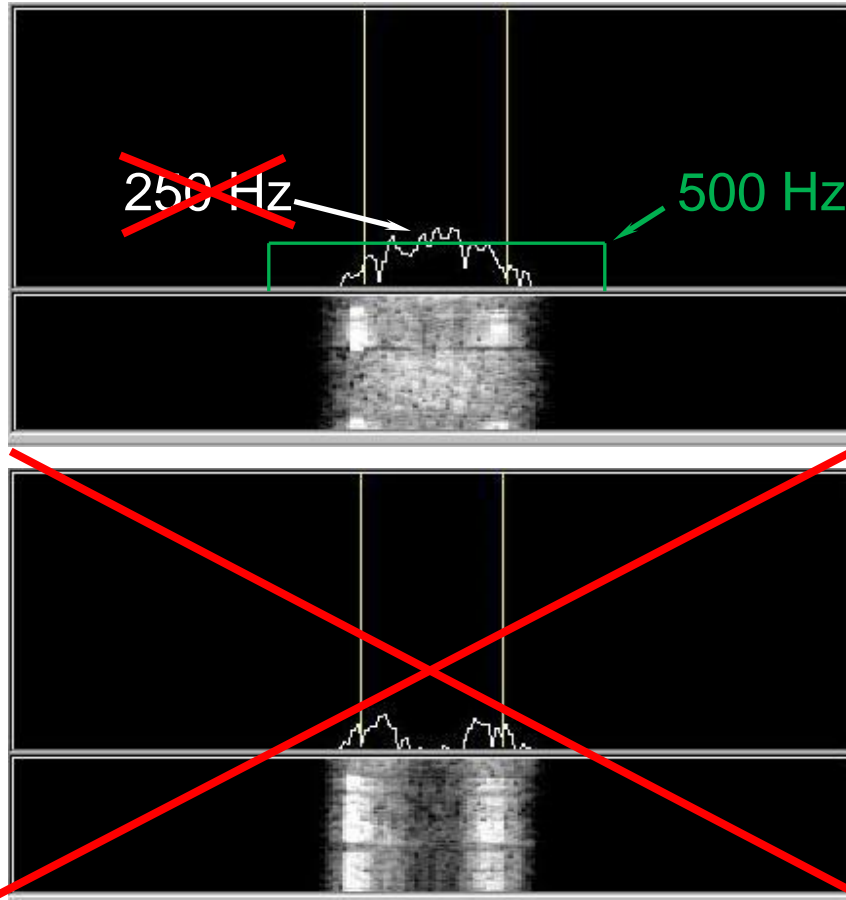
- TX bandwidth
- RX bandwidth
- UOS and hyphen
- Multiple decoders
- SO2V
- SO2R

● FT4/8 Contesting

- CQ vs. S&P mode
- FT8/4 & even/odd
- Working non-contesters
- Superfluous 2nd QSL

RTTY Receive Bandwidth

radio IF filtering



Narrow IF filters

- 500 Hz - normal
- 250 Hz - extreme QRM
- Tone filters – **don't use!**
 - Icom Twin Peak Filter
 - K3 Dual-Tone Filter

RTTY Transmit Bandwidth

unnecessary QRM



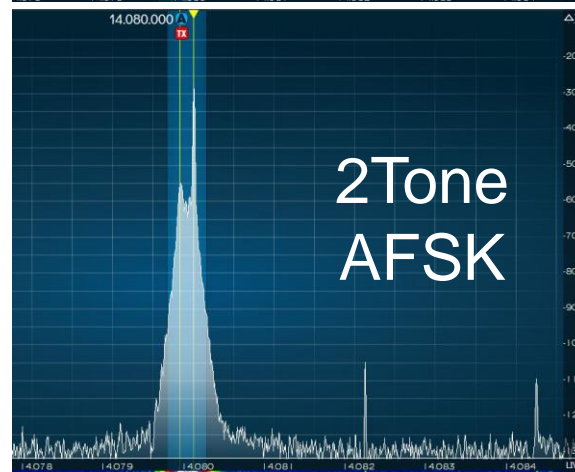
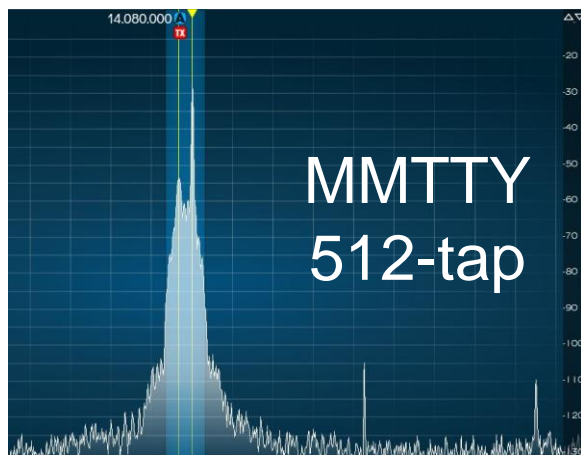
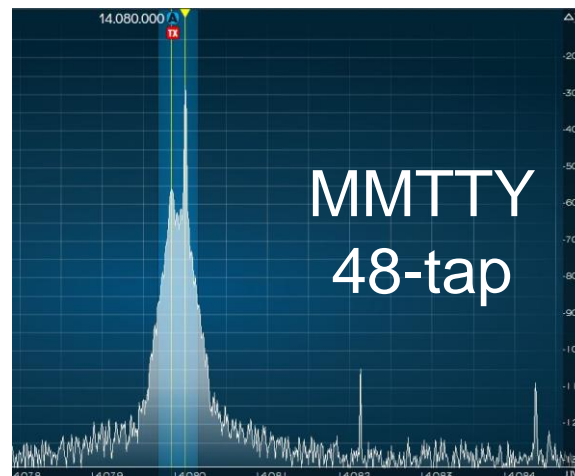
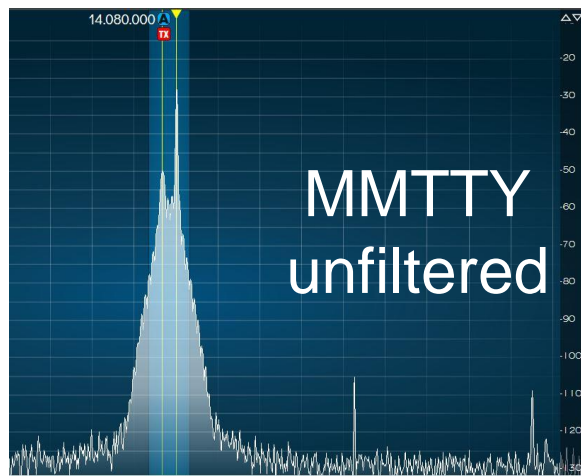
- Wasted power outside receiving decoder BW
 - Suitably narrow TX BW effectively amplifies signal
- Unnecessary QRM
 - Wide 1.5 KW RTTY can QRM 5-10 channels
 - Similar to CW key click problem of the past

Why hurt yourself AND QRM close-by stations?

RTTY Transmit Bandwidth



AFSK



o CTU o

**CONTEST
UNIVERSITY**

Tx BPF Setting

MMTTY



default 48



select 512

W0YK (W0YK) - MMTTY Ver1.70K [based on 12000Hz]

File(F) Edit(E) View(V) Option(O) Profiles(S) Program(P) Help(H)

Control Demodulator (IIR) Macro

FIG Mark 1275

UOS Shift 170

TX BW 60

TXOFF AV. 70

QSO Data Init Call

Setup MMTTY Ver1.70K

Demodulator AFC/ATC/PLL Decode TX Font/Window Misc SoundCard

DIDDLE

NONE

BLK

LTR

Random

WaitTimer

TX

UOS

Double shift

Disable Wait

Disable Rev

Always fix shift

Digital Output

Char. Wait Diddle Wait

PTT & FSK

Port NONE

Invert Logic

Radio command

TxBPF/TxLPF

Tx BPF Tap 48 f

Tx LPF Freq 80 Hz

96

128

144

192

256

386

512

Input Button

1X1 DEAR BTU

Macro

Your Callsign

W0YK 1X2 QANS SK RY

2X3 M6 EE M14

DE3 M7 M11 CQ2

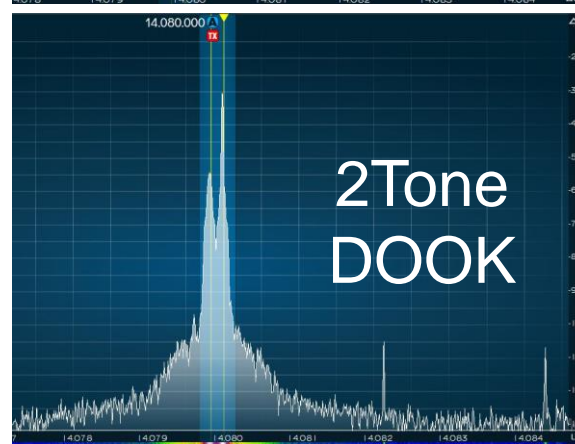
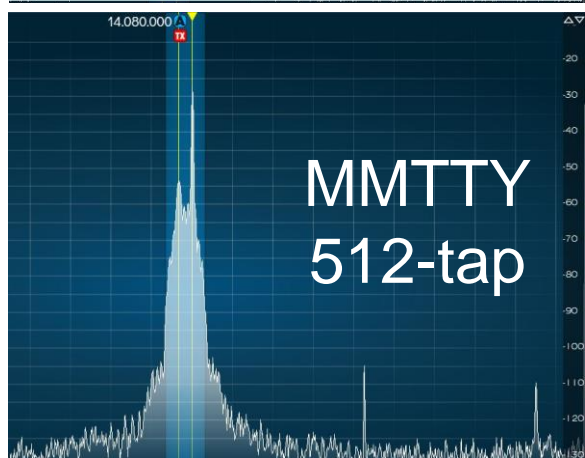
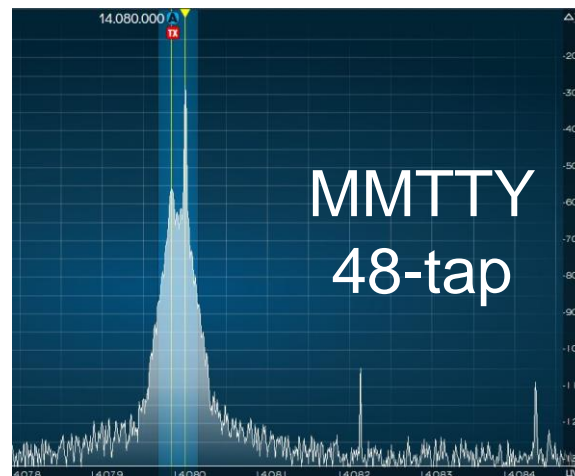
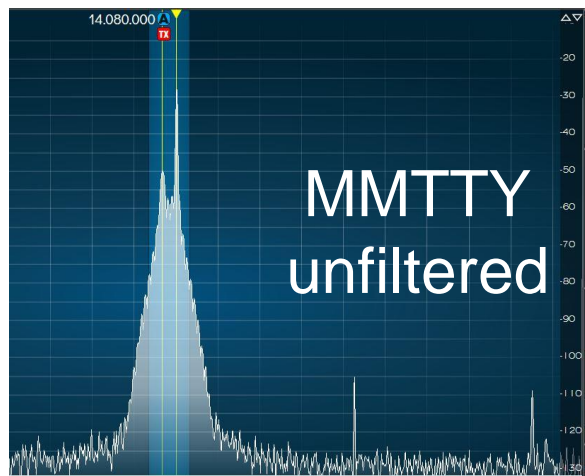
UR599 M8 M12 CQ1

Convert Immediately

HAM Set Default(Demodulator) ? OK Cancel

RTTY Transmit Bandwidth

AFSK – 2Tone DOOK

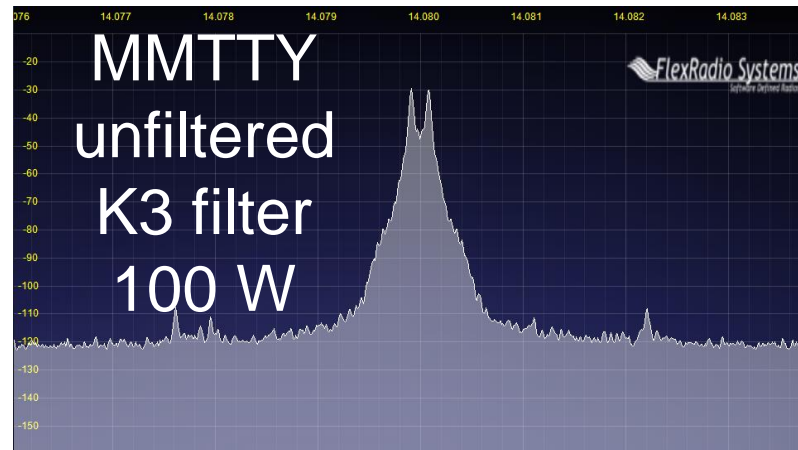
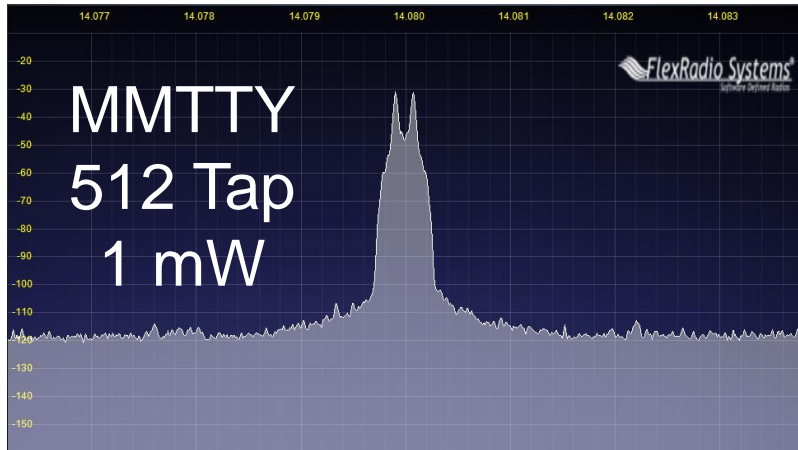


o CTU o

CONTEST
UNIVERSITY

RTTY Transmit Bandwidth

AFSK - PA IMD effect



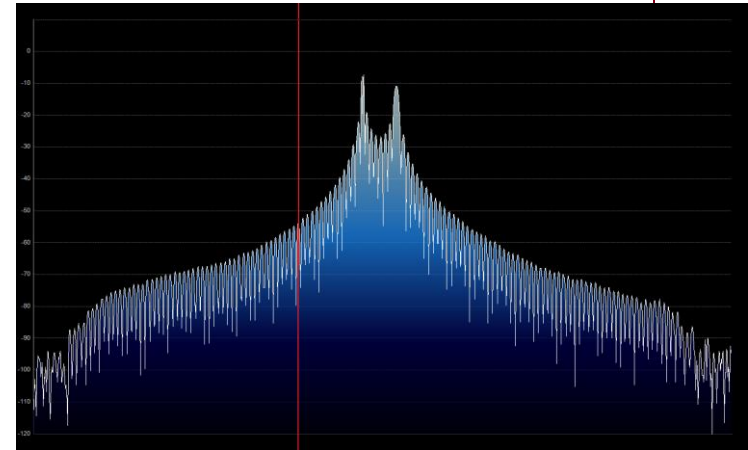
RTTY Transmit Bandwidth

FSK



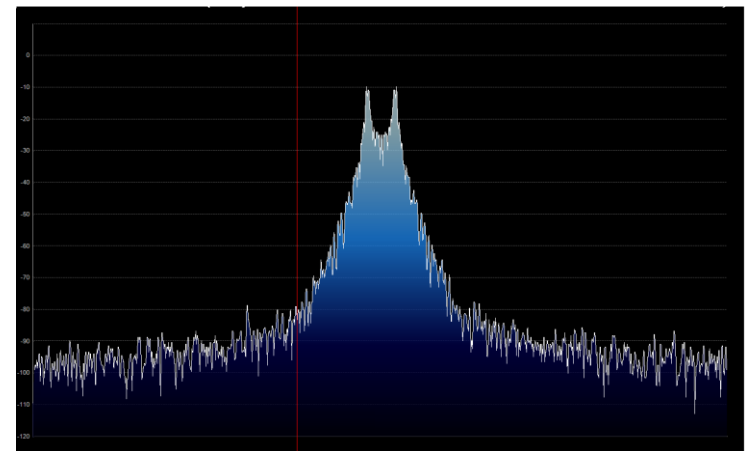
- Old K3 FSK bandwidth

- No waveshaping
- < DSP281 firmware
- Typical of all radios
- 50 watts



- New K3 FSK bandwidth

- Optimal DSP filter
- DSP281 firmware, March 2013



UOS

(Unshift-On-Space)



- Receive UOS:
 - Space character forces a shift to the Letters set
 - Increases noise immunity for alpha text
- Transmit UOS:
 - Sends FIGS character after Space, before numeric “word”
- Contest exchanges are alpha and numeric
 - Should UOS be on or off?
 - Should Space or Hyphen delimit exchange elements?
 - 599 1234 1234 or 599-1234-1234
- *Recommendation:*
 - *Turn on both RX & TX UOS and use Space delimiters*

◦ CTU ◦

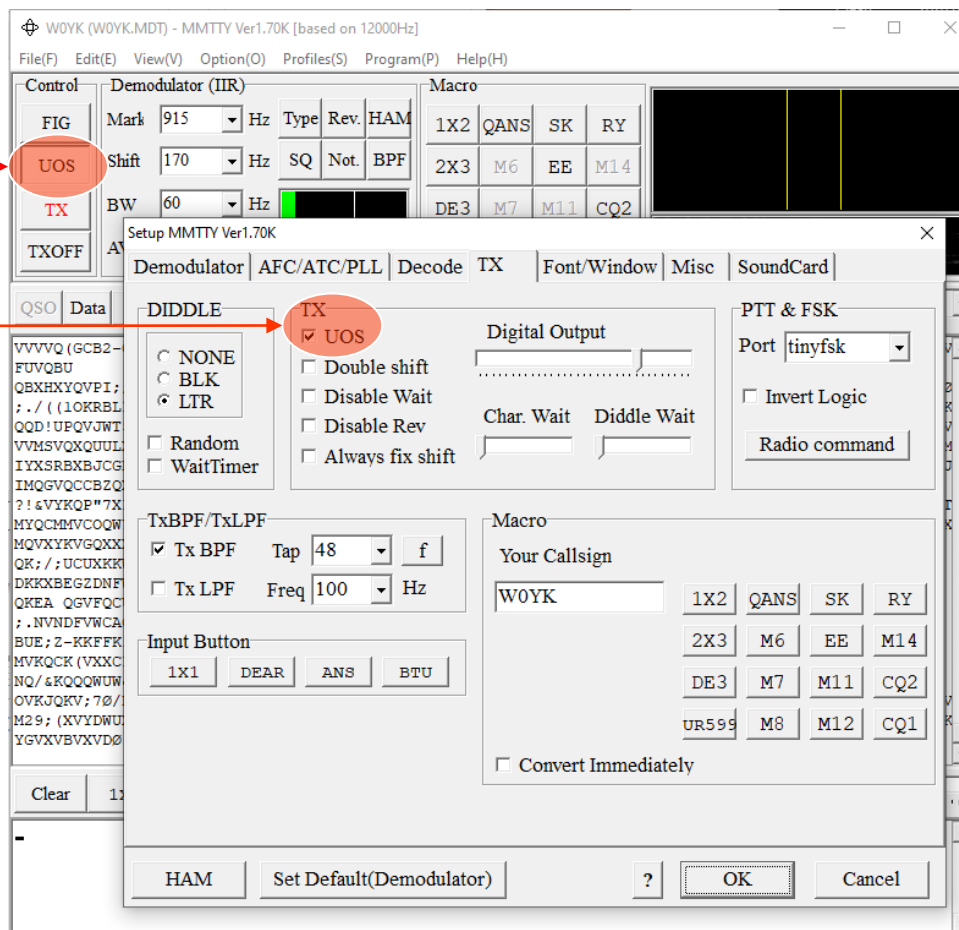
UOS

MMTTY



RX

TX



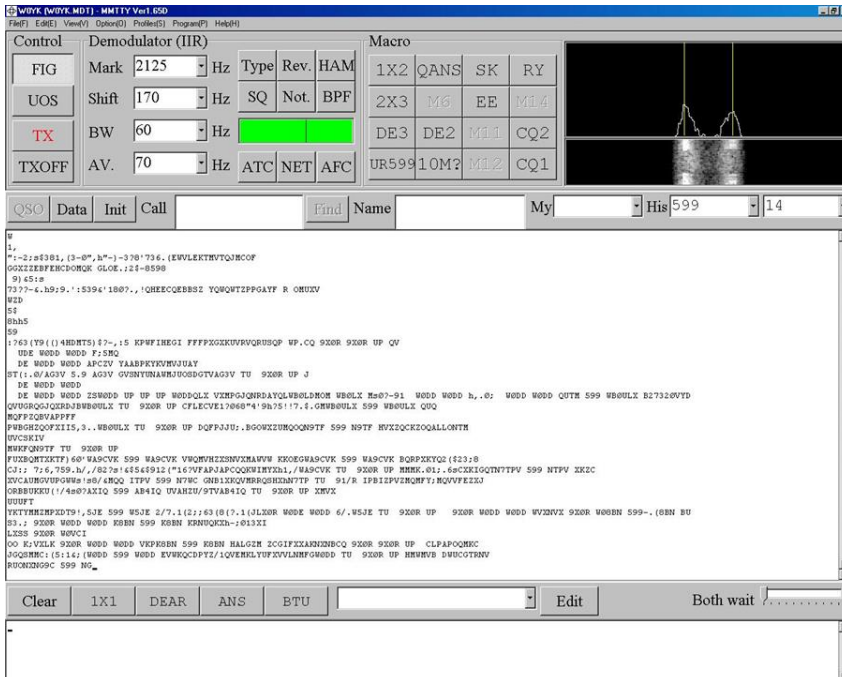
Multiple Decoders



- Parallel decoding with
 - Different decoders
 - Different decoder “profiles”
 - Different RX IF bandwidths (dual receivers)
- Reduces repeats
- Almost “free”
 - Screen space for multiple decoder windows
 - Can be relatively small
 - CPU performance

Multiple Decoders

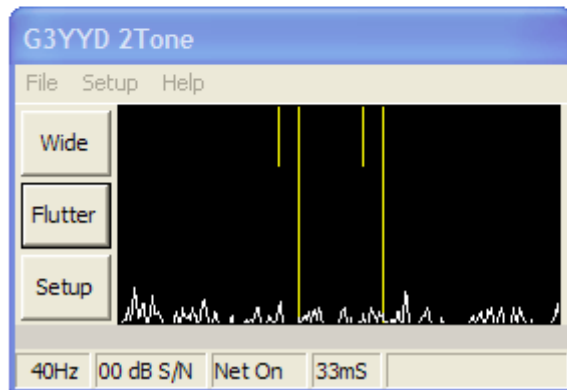
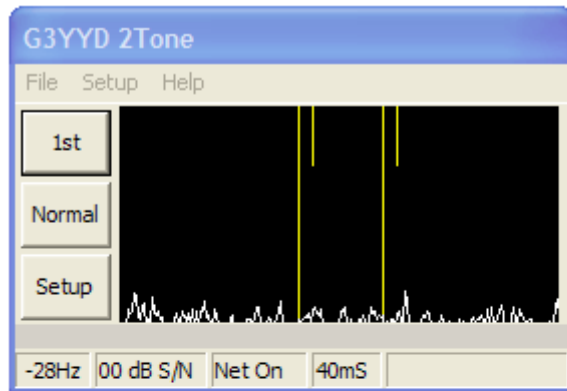
MMTTY



- Dominant SC MODEM
- Standalone, or ...
- Contest loggers:
 - N1MM Logger+
 - WriteLog
 - Win-Test
- Introduced June 2000
- Mako Mori, JE3HHT

Multiple Decoders

2Tone



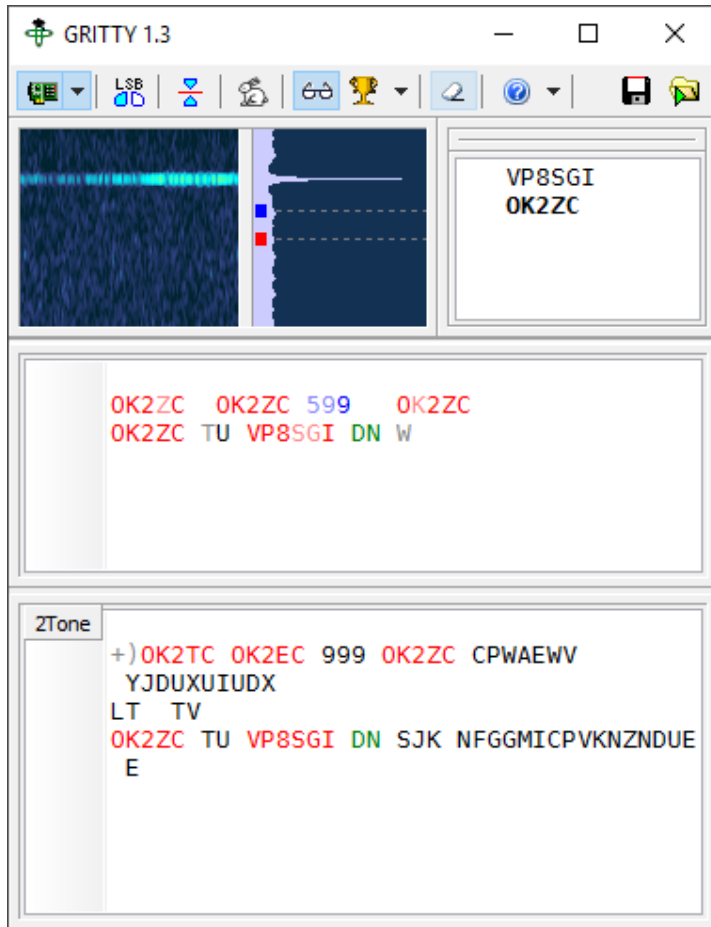
- Outperforms MMTTY ?
- Uses less CPU cycles
- Contest loggers:
 - N1MM Logger+
 - WriteLog
 - Win-Test
- Introduced late 2012
- David Wicks, G3YYD

o CTU o

CONTEST
UNIVERSITY

Multiple Decoders

GRITTY



- Best accuracy ?
- Bayesian statistics
- Standalone, or ...
- Contest loggers:
 - N1MM Logger+ only
- Introduced late 2015
- Alex Shovkopylas, VE3NEA

o CTU o

CONTEST
UNIVERSITY

15/46
18 May 2023

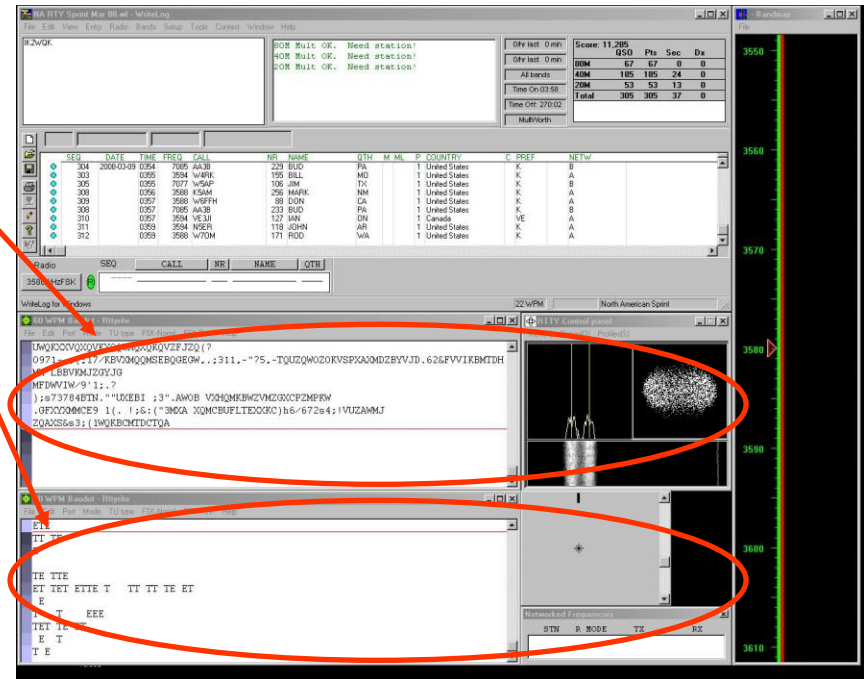
ICOM[®]

Multiple Decoders

MMTTY & DXF38



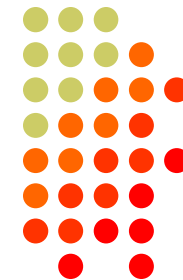
- Parallel decoding
 - Software, e.g., MMTTY
 - Hardware, e.g., DXF38
- Diverse conditions
 - Flutter
 - Multi-path
 - QRM, QRN
 - Weak signals
 - Off-frequency stations



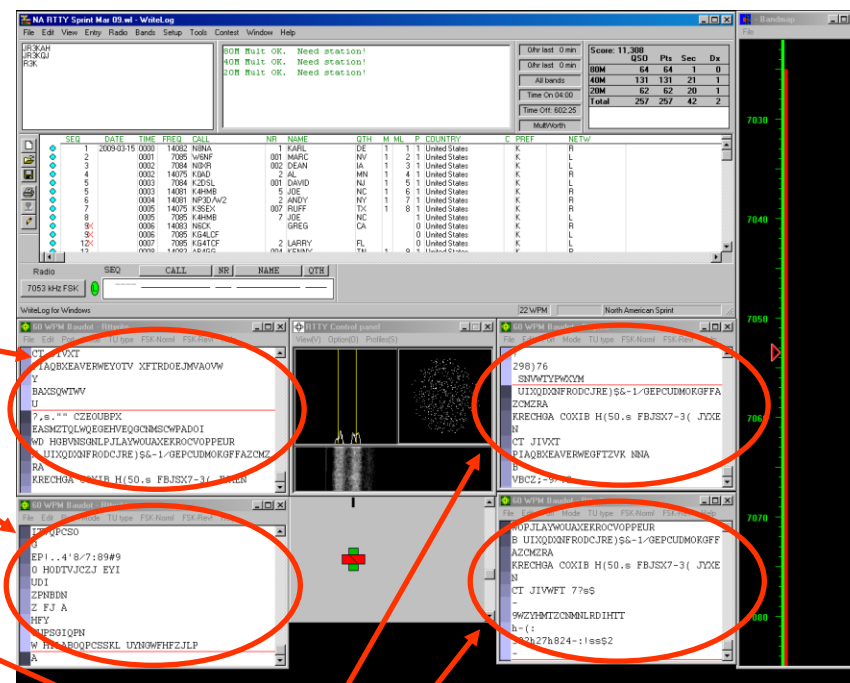
o CTU o

Multiple RTTY Decoders

multiple MMTTY profiles

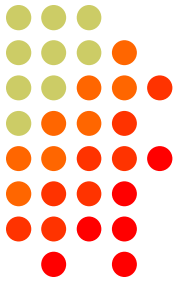


- Parallel decoding
 - same audio stream
 - switching takes too long
- Multiple profile windows
 - Standard
 - Fluttered signals
 - Fluttered signals (FIR)
 - Multi-path
 - hyper sensitive
 - EU1SA
 - AA6YQ-FIR-512
 - weak signals in QRN

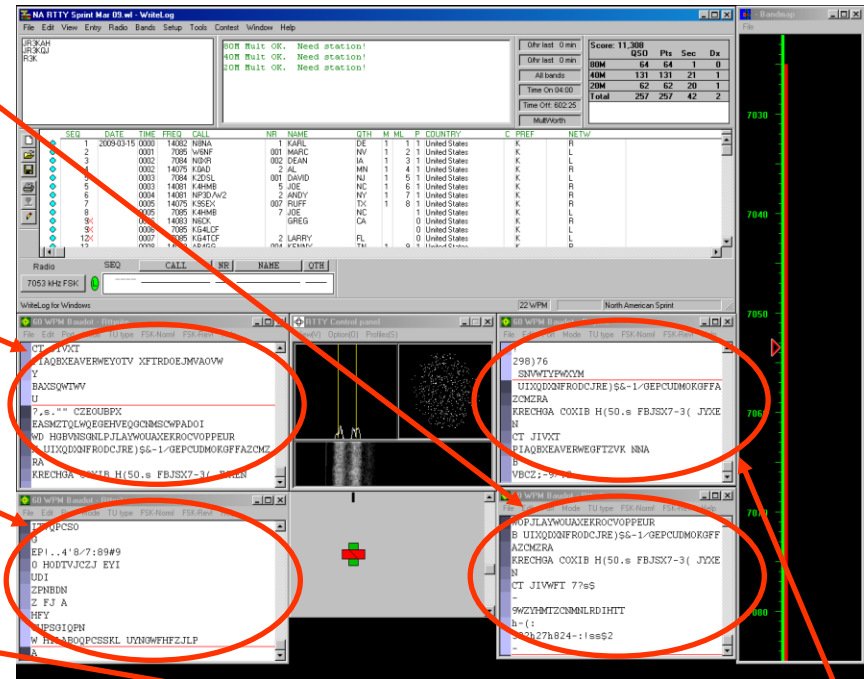


Multiple Decoders

two IF bandwidths



- Narrow IF filtering (main RX)
 - Hardware modem, i.e. DXP38
 - MMTTY profiles:
 - Standard
 - Fluttered signals
 - Fluttered signals (FIR)
 - Multi-path
 - hyper sensitive
 - EU1SA
- Wide IF filtering (sub RX)
 - MMTTY profile:
 - AA6YQ-FIR-512
 - Dual Peak Filter
 - "Matched filter"

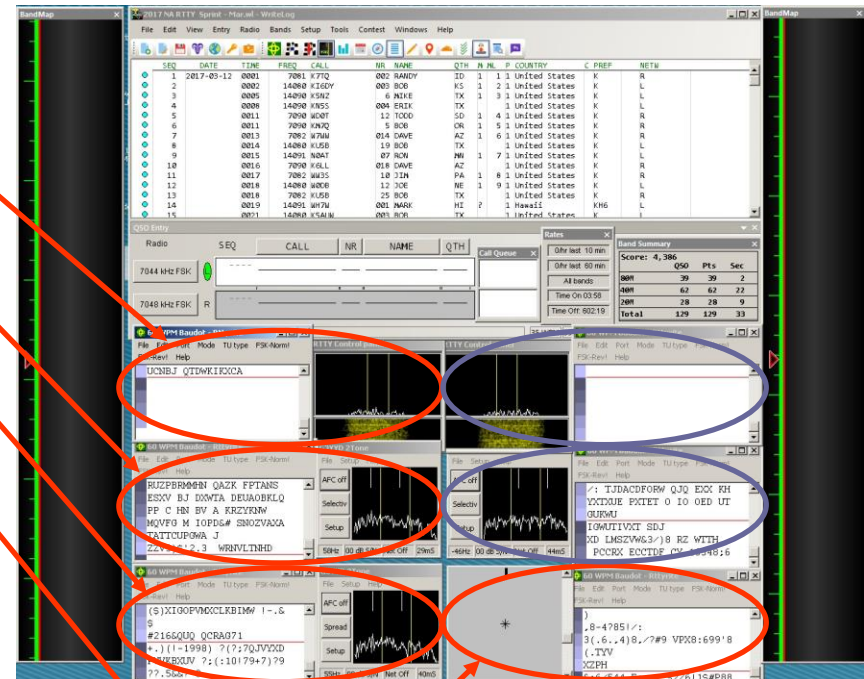


Multiple Decoders

SO2V



- VFO-A (main RX)
 - MMTTY Standard profile
 - 2Tone Flutter profile
 - 2Tone Selective profile
 - DXP38
- VFO-B (sub RX)
 - MMTTY Standard profile
 - 2Tone Flutter profile
- 6 decoders
 - A→B



Multiple Decoders

Tone choices for monitoring



- Low tones are less fatiguing
 - Use high tones for secondary audio stream(s)
- Low/High tones can be mixed to put two audio streams in one ear:
 - SO2R plus SO2V per radio (4 streams)
 - SOnR (3+ streams)

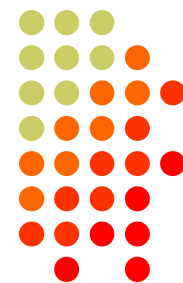
SO2V

“1BSIQ”



1. [single rcvr] If Assisted and running on VFO-A, then
 - A<>B, click spot, tune, ID station, work station
 - A<>B, resume running
2. [dual rcvr] Set up decoder windows on VFO-A and VFO-B
 - Radio must have two true receivers
 - Monitor both frequencies simultaneously with right/left channels of sound card and separate RTTY windows
 - Left-click call from 2nd RTTY window into VFO-B Entry Window
 - Two ways to transmit on VFO-B:
 1. A<>B, work the mult, A<>B (*but, mixes print from two frequencies*)
 2. SPLIT, work the mult, un-SPLIT, resume running
 - Requires “wire-OR’d” FSK or AFSK and two transmit RTTY windows
 - WriteLog **Shared Com Port** obviates the wire-OR
 - K3/WriteLog invokes SPLIT when VFO-B call is clicked

SO2R

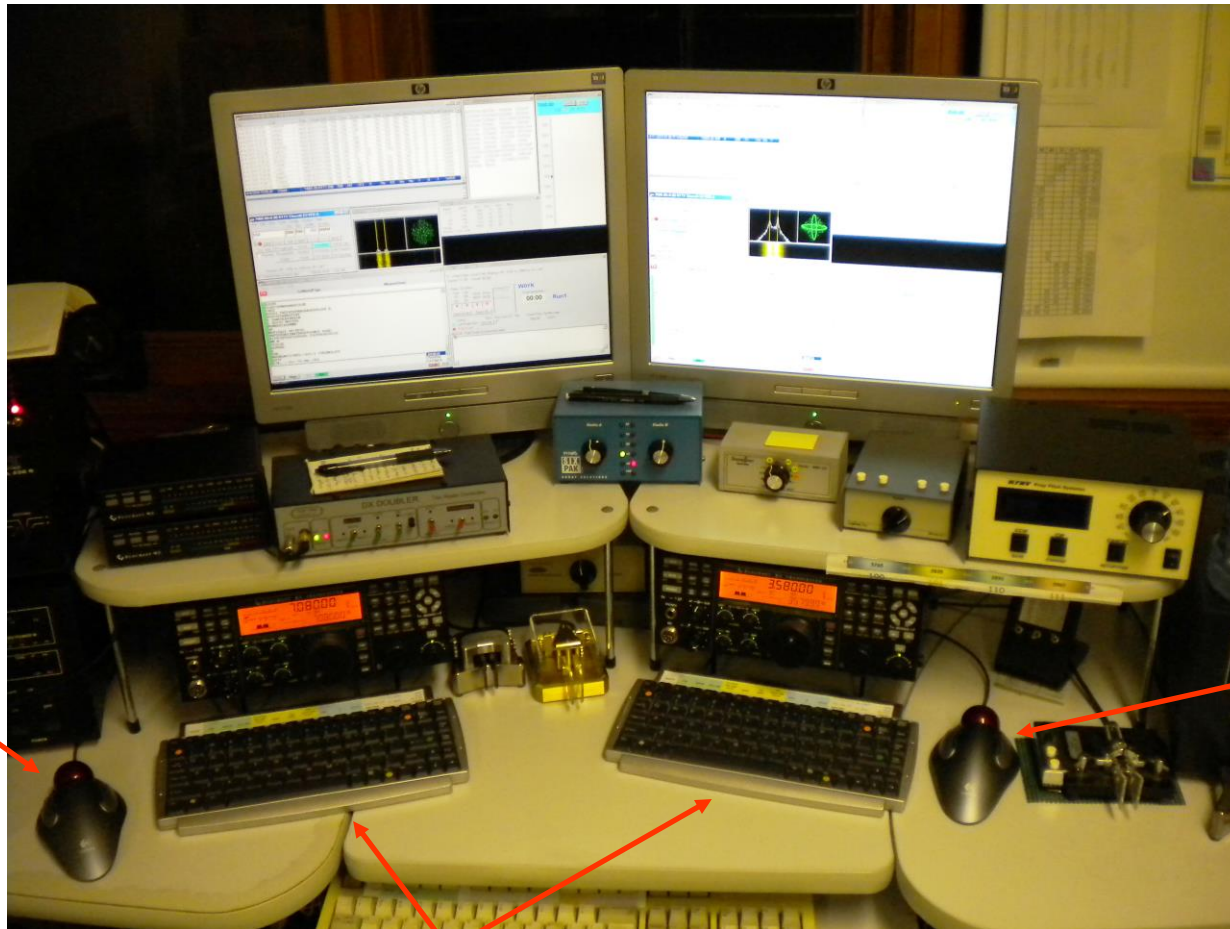


- Eliminates SO1R RTTY “boredom”
- Think beyond run and S&P:
 - Dueling CQs; run on two bands simultaneously (**2BSIQ**)
 - S&P on two bands simultaneously, esp. w/Packet
 - SO2V on one or both radios (SO4V!)
- [optional] Two networked computers:
 - Eliminates swapping radio-focus
 - Display room for more decoder windows per radio
 - RTTY doesn't require much typing; mini-keyboards
 - 2 x SO2V=SO4V for picking up mults on both run bands
 - Easily extendible to SOnR

No time to watch TV or read spy novels!

SO2R

“M2” configuration



Left-hand
Trackball

Right-hand
Trackball

Right-sized
Keyboards

23/46
18 May 2023

SO2R in the NA Sprint

maximize TX duty cycle



- Set VFOs at least 10 kHz apart on both radios
- Find a clear spot on one radio and CQ while you tune the other radio for a station to work
- If you don't find a station to work before the CQ finishes, find a clear frequency and duel CQ
- After a QSO, swap VFOs on that radio, search during other transmission, then resume dueling CQ
- Don't waste time trying to work the "couplet" ... CQing is OK in Sprint!

SOnR

> 2 radios



- Simplify antenna/filter band-decoding:
 - Dedicate a band/antenna to the 3rd (or 4th) radio
- Networked PC/radio simplifies configuration
- RTTY (vs. CW or SSB) easier for operator
 - PC decodes for operator
 - Low tones & high tones allows two radios per ear
 - Classic audio headphone mixer (per ear) provides radio A, radio B or both

SO nR



Multi-Multi configuration



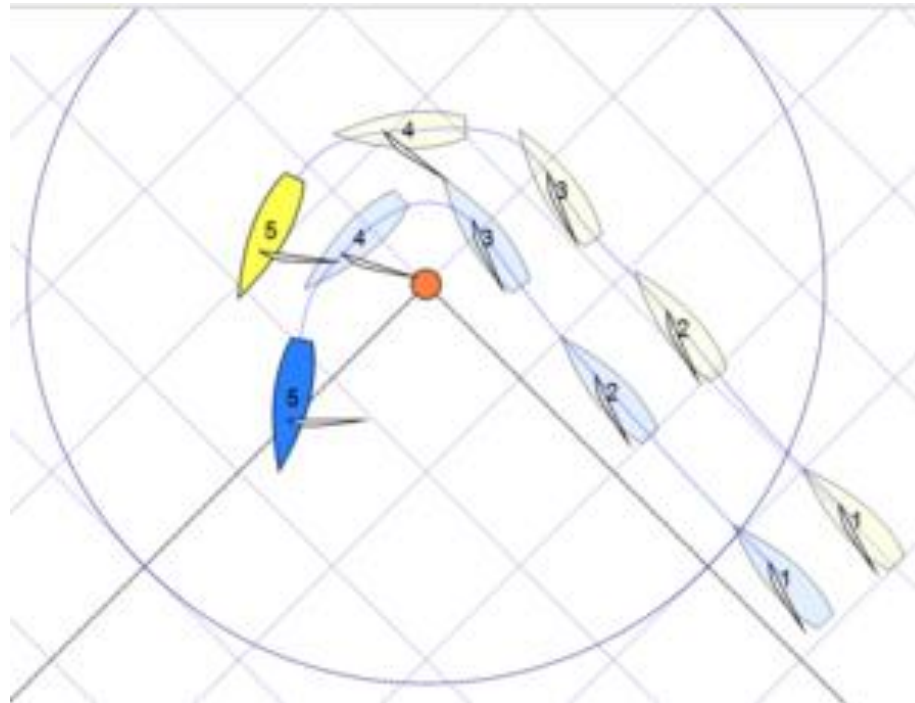
dedicated
to 10 meters

Sailboat Bouy Racing

mark rounding



Yellow falls behind by keeping up with Blue



Call Sign Stacking

“Slow Down to Win”



- Sailboat racing analogy:
 - Pinwheel effect at mark-rounding → slow down
- Let pile-up continue a “beat” after getting the first call sign
 - Increase chance for another call sign or two
 - Increase chance for QSO-phase-skip
- Apply same tactic for tail-enders ... pause ½-second before sending TU/CQ message

Call Sign Stacking

The 4 Phases of a QSO



Normal Run mode flow:

1. CQ msg
 - repeat
 - AGN?
2. pile-up
3. Exchange msg
 - Send fill(s)
4. receive his Exchange
 - AGN? or NR? or QTH? or NAME?

1. TU/CQ msg (logs QSO)

Normal S&P mode flow:

1. CQ
2. <mycall> msg
 - repeat
3. receive his Exchange
 - AGN? or NR? or QTH? or NAME?
4. Exchange msg
 - send fill(s)

1. find next CQ

transmit

receive

Call Sign Stacking

Pileup



Normal

1. WPX P49X P49X CQ, or
TU P49X CQ
2. K3LR K3LR K5ZD K5ZD
3. K3LR 599 2419 2419
4. TU 599 842 842

Shortened

1. (skip CQ)
2. (skip pileup)
3. K3LR TU NW
K5ZD 599 2420 2420
4. TU 599 1134 1134

transmit

receive

Call Sign Stacking

Tail-end



Normal

1. WPX P49X P49X CQ, or
TU P49X CQ
2. K3LR K3LR
3. K3LR 599 2419 2419
K5ZD (*tail-end*)
4. TU 599 842 842

Shortened

1. (skip CQ)
2. (skip pileup)
3. K3LR TU NW
K5ZD 599 2420 2420
4. TU 599 1134 1134

transmit

receive

Call Sign Stacking

summary



- Efficiently work:
 - multiple callers in a pile-up, and
 - tail-enders to a completing QSO
- Calls **pushed** onto the stack as they arrive
- Message parameter **pops** call off of the stack into the Entry window
- Eliminates 2 of 4 QSO phases, which doubles short-term rate

FT8 Multi-Channel Reception

Run vs. S&P is irrelevant



02:34:45

02:35:15

02:34:45

02:36:15

My Tx

His Tx

Rotate Odd/Even Cycles



The screenshot shows the WSJT-X v2.2.0-rc1 interface. The 'Monitor' button is highlighted with a red arrow. The 'Tx even/1st' checkbox is checked. The frequency is set to 14.074 000. The interface also shows a 'Generate Std Msgs' table with the following entries:

Generate Std Msgs	Next	Now	Pwr
AA5AU W0YK CM97	<input type="radio"/>	Tx 1	
AA5AU W0YK CM97	<input type="radio"/>	Tx 2	
AA5AU W0YK R CM97	<input type="radio"/>	Tx 3	
AA5AU W0YK RR73	<input type="radio"/>	Tx 4	
AA5AU W0YK 73	<input type="radio"/>	Tx 5	
CQ WW W0YK CM97	<input checked="" type="radio"/>	Tx 6	

Rotate FT4/FT8 Modes



The screenshot shows the WSJT-X v2.2.0 software window. The 'View' menu is open, and the 'Mode' option is highlighted with a red circle. The main interface shows a frequency of 14.074 000 MHz. The 'Generate Std Msgs' panel is visible, showing a list of messages for rotation:

Message	Next	Now
AA5AU W0YK CM97	<input type="radio"/>	Tx 1
AA5AU W0YK CM97	<input type="radio"/>	Tx 2
AA5AU W0YK R CM97	<input type="radio"/>	Tx 3
AA5AU W0YK RR73	<input type="radio"/>	Tx 4
AA5AU W0YK 73	<input type="radio"/>	Tx 5
CQ WW W0YK CM97	<input checked="" type="radio"/>	Tx 6

The status bar at the bottom indicates 'Receiving' and 'FT8' mode.

Working Non-Contesters



- Depends on contest
 - Grid Square exchange
 - QTH, serial number, name, etc.
- Transparant ... unless
 - Non-contester skips Tx2, answering with Tx3
- ***Recommendation: Don't call CQ, only answer CQs or messages with Grid Square***

Gangsters Problem ^[1]

unreliable communication



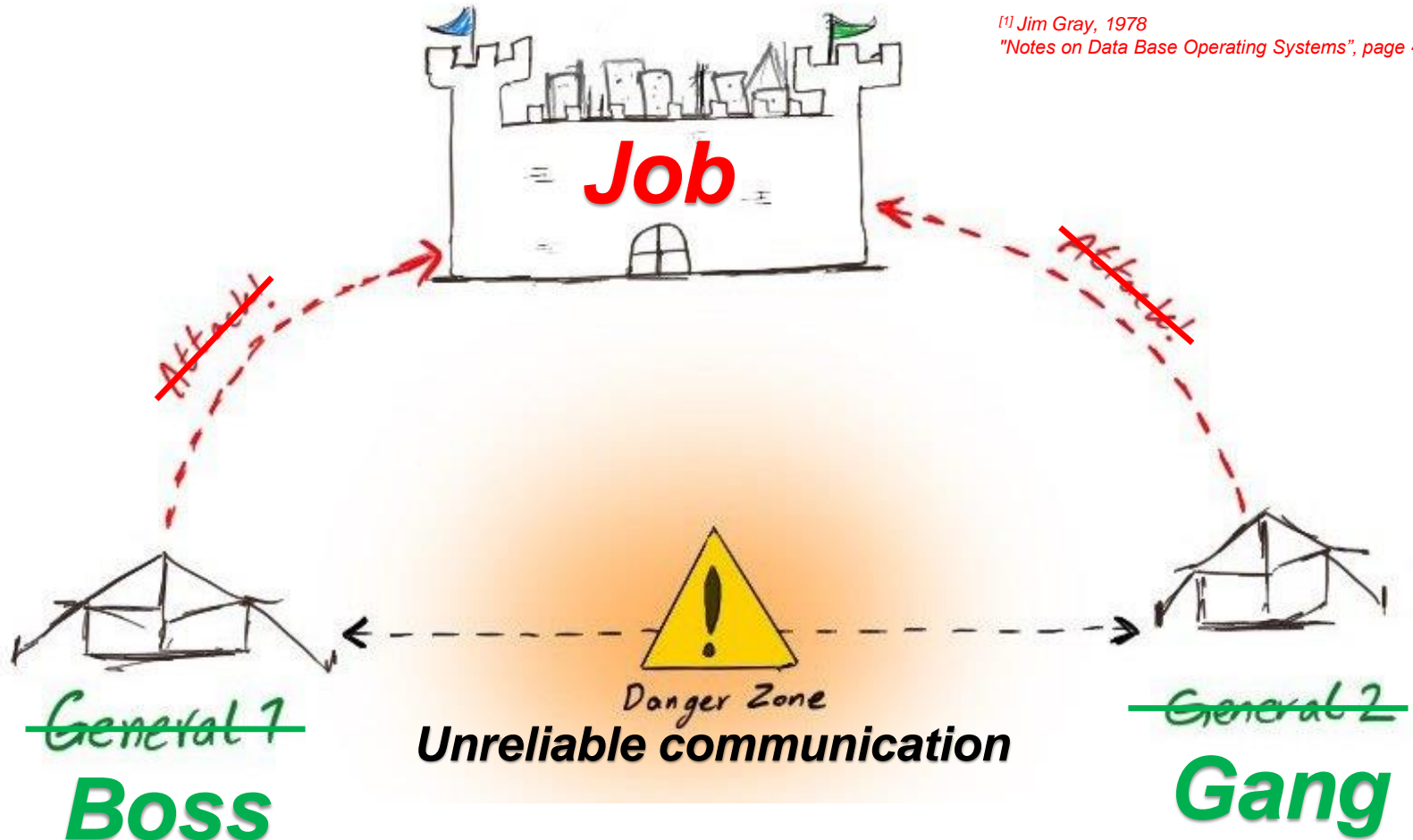
^[1] E. A. Akkoyunlu, K. Ekanadham, and R. V. Huber, 1975
"Some Constraints and Trade-offs in the Design of Network Communications", page 73

- 1975 computer science thought experiment
- Communication over an unreliable link
 - e.g., TCP
- ACKs could theoretically be infinite
- Solution
 - Accept some uncertainty; don't try to eliminate
 - Mitigate to reduce consequence(s)

~~Two Generals Paradox~~ [1] The Gangsters



[1] Jim Gray, 1978
"Notes on Data Base Operating Systems", page 465



Radiosport Solution

CW, SSB & RTTY



- Each QSO partner QSLs the exchange *once*
- Context reduces uncertainty
 - Other station doesn't repeat their last message
 - Other station doesn't ask for a repeat
 - Other station rolls into their next QSO

Radiosport Solution

FT4 & FT8



- One QSO partner QSLs the **QSL**
- Implied by default WSJT-X logging behavior
- Defacto expectation
 - Many FT ops won't log the QSO without this superfluous QSL of the final QSL
 - Thus, NIL rate increases
 - CW, SSB & RTTY = 1-2%
 - FT = 5-6%

WW Digi QSO



CQ W0YK CM97

W0YK AA5AU EL92 ← AA5AU calls with exch

AA5AU W0YK R CM97 ← W0YK QSL with exch

W0YK AA5AU RR73 ← AA5AU QSL

AA5AU W0YK 73 ← W0YK QSLs AA5AU's QSL!

*This wastes time because W0YK could
have used the message to CQ or
answer another caller.*

WW Digi Alternative QSO

context



CQ W0YK CM97

W0YK AA5AU EL92

← AA5AU calls with exch

AA5AU W0YK R CM97

← W0YK QSL's with exch

W0YK AA5AU RR73

← AA5AU QSL's

CQ W0YK CM97

← W0YK calls CQ,

or

AC0C W0YK R CM97

← W0YK rolls into next QSO

*AA5AU then knows, by context,
that W0YK received his QSL message*

WW Digi Alternative QSO

message repeat



CQ W0YK CM97

W0YK AA5AU EL92

AA5AU W0YK R CM97

W0YK AA5AU RR73

AA5AU W0YK R CM97

W0YK AA5AU RR73

←AA5AU calls with exch

← W0YK QSL's with exch

←AA5AU QSL's

← W0YK missed QSL msg

←AA5AU repeats QSL

Minimizing NILs

Recommendation #1



- Develop skill to dynamically change message
 - e.g., use the Alternate F1-F6 keys in WSJT-X
- Always log the QSO when receiving a RRR, RR73 or 73 message.
- Always log the QSO when sending RRR, RR73 or 73 message.
 - Look for a clue that your message was not received, e.g., your QSO partner re-sends his report.

Minimizing NILs

Recommendation #2



- Give in!
 - Send the superfluous 73, but
 - Don't require it from your QSO partner
- Yes, it's unnecessarily slower, but
 - FT contesting is currently slow enough to absorb it



ROOM 3 – Digital and RTTY Contesting – W0YK

3:15 OPEN DISCUSSION Q&A