Tweaking ECP Performance: A libRIST Deep Dive

libRIST

Sergio Ammirata, PhD.
Chief Scientist, SipRadius
& Maintainer, libRIST

March 1, 2021
Virtual VidTrans 21

VSF
VIDEO SERVICES FORUM

Tweaking ECP Performance:
A libRIST Deep Dive:

Who is SipRadius and what is libRIST and what do you mean by ECP …

ECP = "Error Correction Protocol" = RIST, SRT, DOZER, etc.

SipRadius

libRIST

Who are we:

SipRadius

- Creators and maintainers of Coral-Linux
- 11+ Years creating network transport products for the broadcast industry.
- Entered the ECP arena 7 years ago with the creation of the patented DOZER protocol.

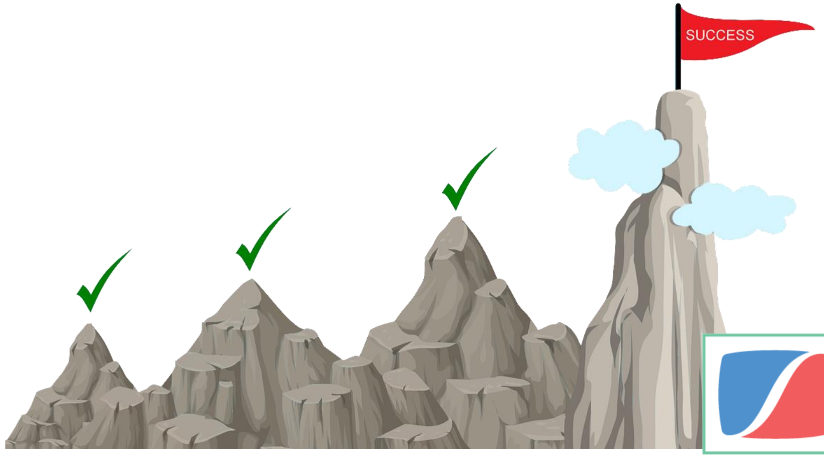
What is libRIST

libRIST

- An open source library used to easily add the RIST protocol to your application or appliance.
- Inherited 7 years of know-how from SipRadius' Dozer protocol.
- Mature, production ready solution with no strings attached (commercial use is allowed and free)

What We'll Discuss Today

+ LibRIST major features
+ Why makes it different that other ECP solutions
+ Live Demo/Stress Test



libRIST, one year later

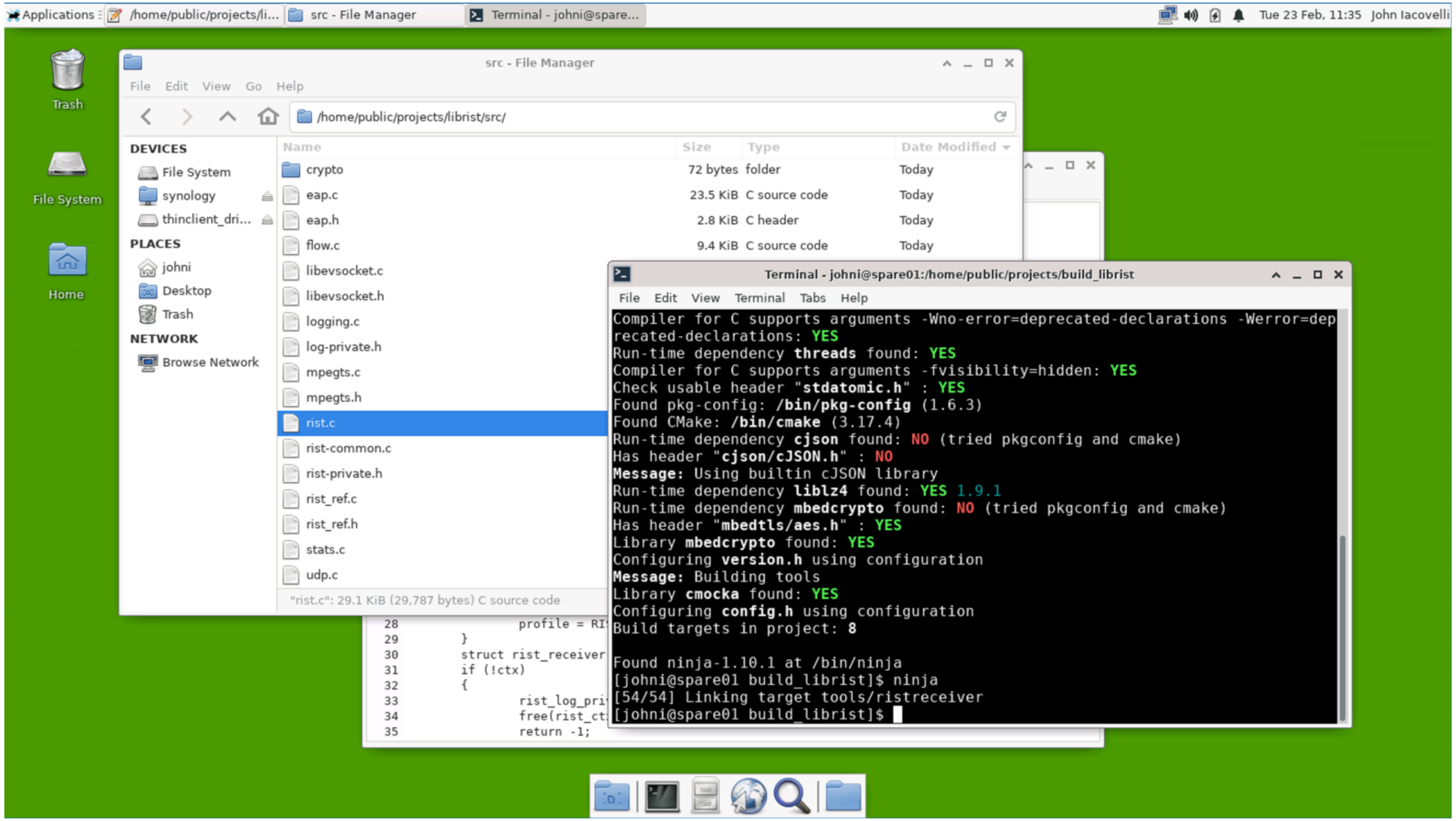
675 commits after initial public release …

- Reaching Most Platforms
  - Intel x64, ARM, AWS, Azure, VMWare
  - linux, Windows, Darwin, IOS, Android
- Performs at levels beyond many commercial protocols.
- Just released v0.2.0, being integrated into VLC 4.0 and ffmpeg 4.3

VSF
VIDEO SERVICES FORUM

Reaching More Platforms

Building libRIST in a linux desktop



Applications
/home/public/projects/li...
src - File Manager
Terminal - johni@spare...
Tue 23 Feb, 11:35 John Iacovelli
Trash
File System
Home
src - File Manager
File Edit View Go Help
/home/public/projects/librist/src/
DEVICES
File System
synology
thinclient_dri...
PLACES
johni
Desktop
Trash
NETWORK
Browse Network
Terminal - johni@spare01:/home/public/projects/build_librist
File Edit View Terminal Tabs Help

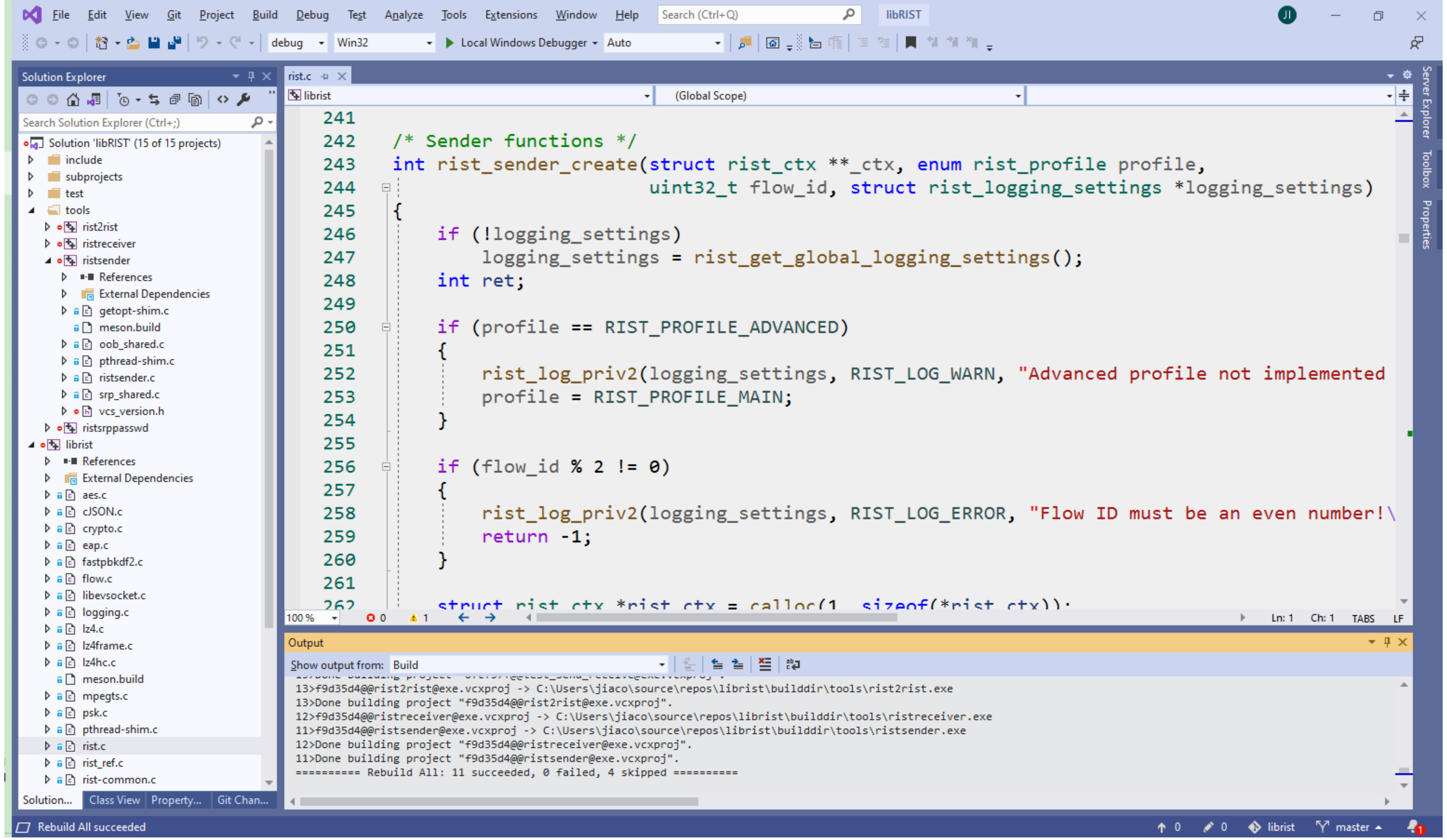
Name	Size	Type	Date Modified
crypto	72 bytes	folder	Today
eap.c	23.5 KiB	C source code	Today
eap.h	2.8 KiB	C header	Today
flow.c	9.4 KiB	C source code	Today
libevsocket.c			
libevsocket.h			
logging.c			
log-private.h			
mpegts.c			
mpegts.h			
rist.c			
rist-common.c			
rist-private.h			
rist_ref.c			
rist_ref.h			
stats.c			
udp.c			

"rist.c": 29.1 KiB (29,787 bytes) C source code

```
Compiler for C supports arguments -Wno-error=deprecated-declarations -Werror=dep
recated-declarations: YES
Run-time dependency threads found: YES
Compiler for C supports arguments -fvisibility=hidden: YES
Check usable header "stdatomic.h" : YES
Found pkg-config: /bin/pkg-config (1.6.3)
Found CMake: /bin/cmake (3.17.4)
Run-time dependency cjson found: NO (tried pkgconfig and cmake)
Has header "cjson/cJSON.h" : NO
Message: Using builtin cJSON library
Run-time dependency liblz4 found: YES 1.9.1
Run-time dependency mbedcrypto found: NO (tried pkgconfig and cmake)
Has header "mbedtls/aes.h" : YES
Library mbedcrypto found: YES
Configuring version.h using configuration
Message: Building tools
Library cmocka found: YES
Configuring config.h using configuration
Build targets in project: 8

Found ninja-1.10.1 at /bin/ninja
[johni@spare01 build_librist]$ ninja
[54/54] Linking target tools/ristreceiver
[johni@spare01 build_librist]$
```

```
                profile = RI
        }
        struct rist_receiver
        if (!ctx)
        {
                rist_log_pri
                free(rist_ct
                return -1;
```

libRIST in Microsoft Visual Studio

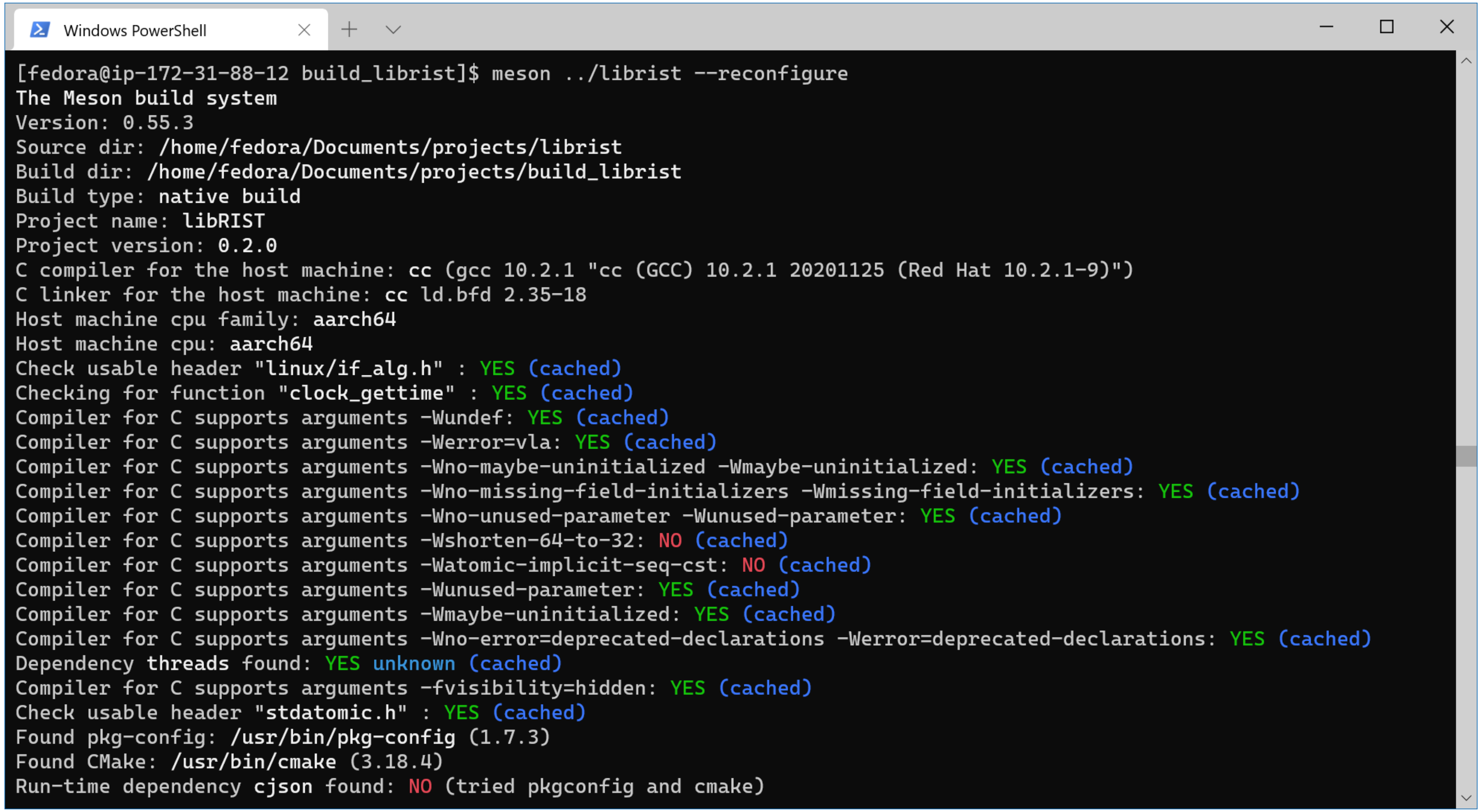


```c

/* Sender functions */
int rist_sender_create(struct rist_ctx **_ctx, enum rist_profile profile,
                       uint32_t flow_id, struct rist_logging_settings *logging_settings)
{
    if (!logging_settings)
        logging_settings = rist_get_global_logging_settings();
    int ret;

    if (profile == RIST_PROFILE_ADVANCED)
    {
        rist_log_priv2(logging_settings, RIST_LOG_WARN, "Advanced profile not implemented
        profile = RIST_PROFILE_MAIN;
    }

    if (flow_id % 2 != 0)
    {
        rist_log_priv2(logging_settings, RIST_LOG_ERROR, "Flow ID must be an even number!\
        return -1;
    }

    struct rist_ctx *rist_ctx = calloc(1, sizeof(*rist_ctx));
```

```
13>f9d35d4@@rist2rist@exe.vcxproj -> C:\Users\jiaco\source\repos\librist\builddir\tools\rist2rist.exe
13>Done building project "f9d35d4@@rist2rist@exe.vcxproj".
12>f9d35d4@@ristreceiver@exe.vcxproj -> C:\Users\jiaco\source\repos\librist\builddir\tools\ristreceiver.exe
11>f9d35d4@@ristsender@exe.vcxproj -> C:\Users\jiaco\source\repos\librist\builddir\tools\ristsender.exe
12>Done building project "f9d35d4@@ristreceiver@exe.vcxproj".
11>Done building project "f9d35d4@@ristsender@exe.vcxproj".
========== Rebuild All: 11 succeeded, 0 failed, 4 skipped ==========
```

Rebuild All succeeded

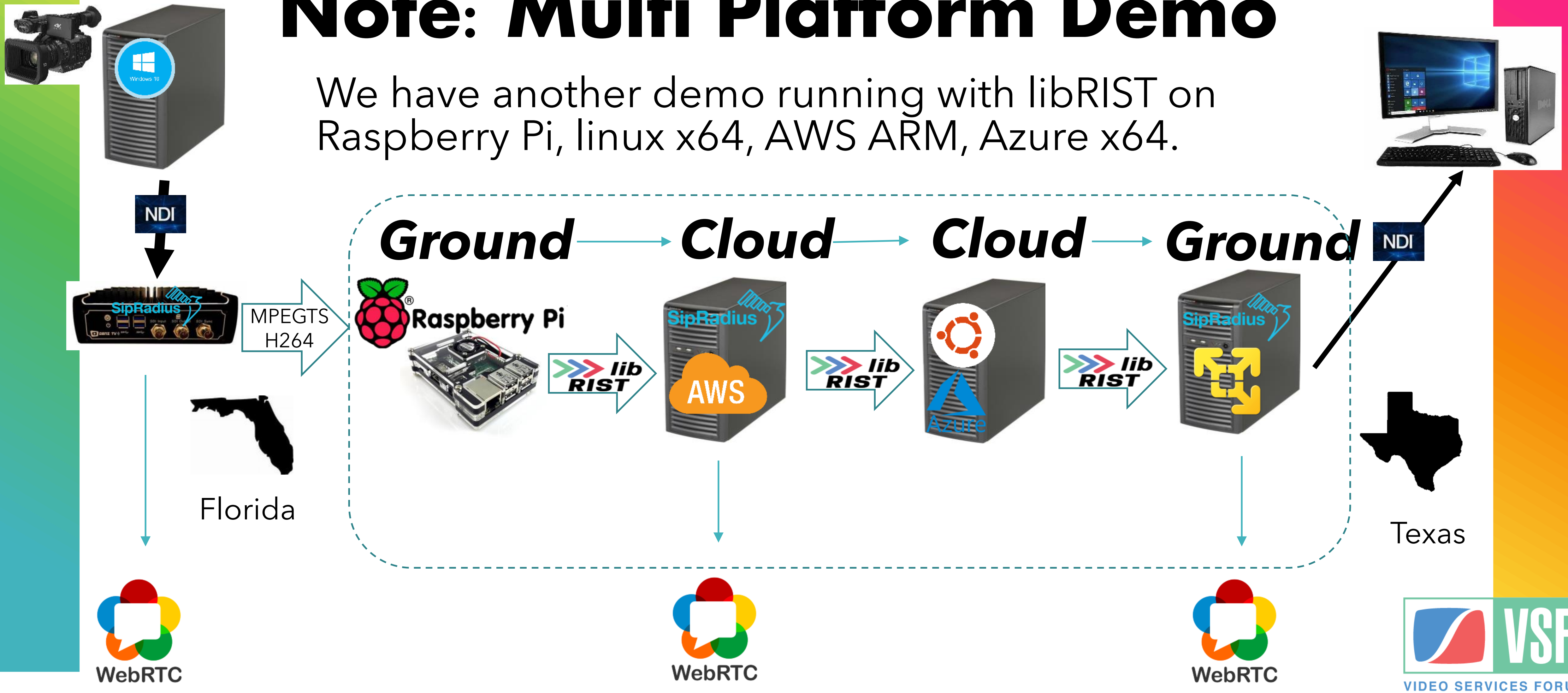
LibRIST in AWS ARM (Gravitron)



```
[fedora@ip-172-31-88-12 build_librist]$ meson ../librist --reconfigure
The Meson build system
Version: 0.55.3
Source dir: /home/fedora/Documents/projects/librist
Build dir: /home/fedora/Documents/projects/build_librist
Build type: native build
Project name: libRIST
Project version: 0.2.0
C compiler for the host machine: cc (gcc 10.2.1 "cc (GCC) 10.2.1 20201125 (Red Hat 10.2.1-9)")
C linker for the host machine: cc ld.bfd 2.35-18
Host machine cpu family: aarch64
Host machine cpu: aarch64
Check usable header "linux/if_alg.h" : YES (cached)
Checking for function "clock_gettime" : YES (cached)
Compiler for C supports arguments -Wundef: YES (cached)
Compiler for C supports arguments -Werror=vla: YES (cached)
Compiler for C supports arguments -Wno-maybe-uninitialized -Wmaybe-uninitialized: YES (cached)
Compiler for C supports arguments -Wno-missing-field-initializers -Wmissing-field-initializers: YES (cached)
Compiler for C supports arguments -Wno-unused-parameter -Wunused-parameter: YES (cached)
Compiler for C supports arguments -Wshorten-64-to-32: NO (cached)
Compiler for C supports arguments -Watomic-implicit-seq-cst: NO (cached)
Compiler for C supports arguments -Wunused-parameter: YES (cached)
Compiler for C supports arguments -Wmaybe-uninitialized: YES (cached)
Compiler for C supports arguments -Wno-error=deprecated-declarations -Werror=deprecated-declarations: YES (cached)
Dependency threads found: YES unknown (cached)
Compiler for C supports arguments -fvisibility=hidden: YES (cached)
Check usable header "stdatomic.h" : YES (cached)
Found pkg-config: /usr/bin/pkg-config (1.7.3)
Found CMake: /usr/bin/cmake (3.18.4)
Run-time dependency cjson found: NO (tried pkgconfig and cmake)
```

Note: Multi Platform Demo

We have another demo running with libRIST on Raspberry Pi, linux x64, AWS ARM, Azure x64.



NDI

SipRadius

MPEGTS H264

Ground → Cloud → Cloud → Ground

Raspberry Pi

lib RIST

SipRadius

AWS

lib RIST

Azure

lib RIST

SipRadius

NDI

Florida

Texas

WebRTC

WebRTC

WebRTC

VSF VIDEO SERVICES FORUM

Performance Gains

Why do we care?

VIDEO SERVICES FORUM

Congestion Control

- Many proprietary ECP solutions exist: Dozer, Zixi, LTN, Video Flow, SRT, NetInsight, etc …
- Many vendors have standarized and implemented RIST: SipRadius, Cobalt, VideoFlow, QVidium, NetInsight, Nevion, etc …
- They all work well under well behaved conditions (low packet loss and unlimited bandwidth)

Congestion Control

+ What happens when the network is not well behaved, i.e. internet links?
  - Bursts of high packet loss > 20%
  - Continuous packet loss of 50% or more (bonded solutions in which one or more legs go down).
  - Total available bandwidth is < 20% above the transmission bitrate
+ This is where the different solutions and implementations diverge

Congestion Control - libRIST

+ Implements internal QOS to control max bandwidth used.

+ Measures and uses round trip time (RTT)

+ Spread the retries over time to prevent network collapse (retry queue).

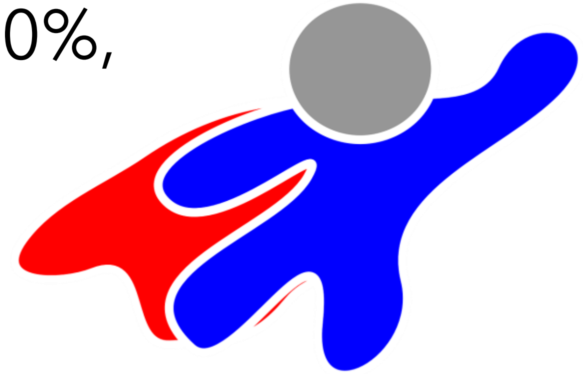
+ Double check data on retry queue before acting on it

Congestion Control - libRIST

- Benefits
  - Less messages mean less congestion
  - Less congestion means more room for new and resent packets
  - QOS means your retry storms will not artificially collapse your links

How Do We Measure Gains?

- Increased ability to withstand random packet loss and corruption under limited bandwidth constraints
  - Previously: no frame drops or corruption up to just under 50% continuous loss
  - Improved performance: up to nearly 70%, which we'll see in the demo, in a short while…



Summarizing the Tweaks

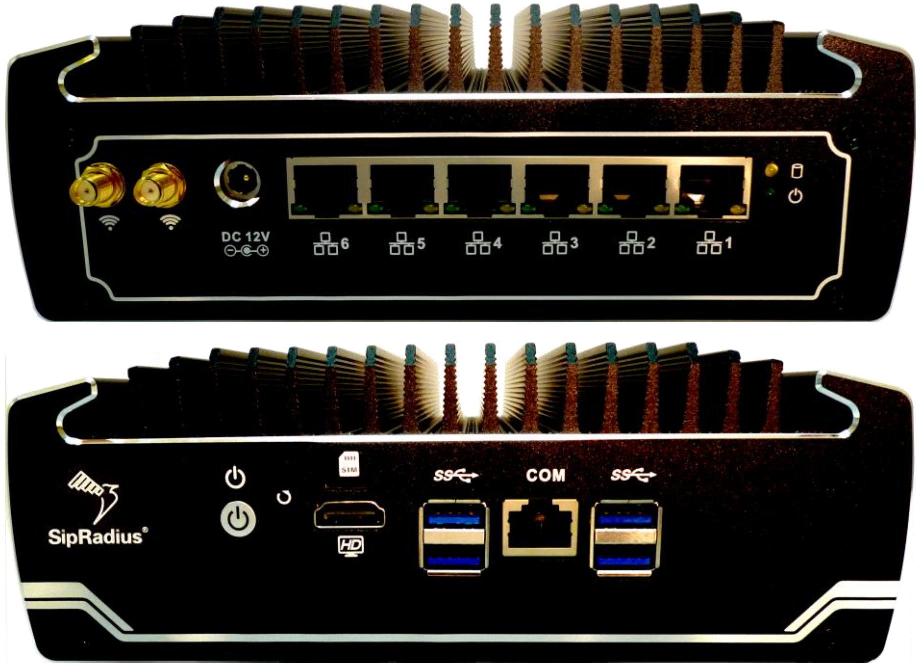
- Most approaches that we know of rely on statistical data being transmitted from client to receiver and vice-versa (RTCP has data for it).
- Our approach is to do it *before* the data aggregation step.
- It may be more work, but the CPU usage for libRIST is so low, there's plenty of leeway.

VIDEO SERVICES FORUM

Demo/Stress Test

SipRadius Emulation Platform

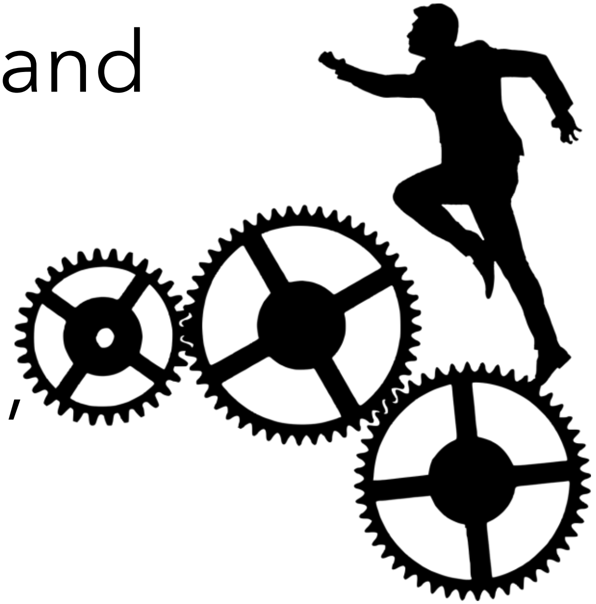
- Core™ i7 Six NIC Fanless Mini PC
- Coral OS linux
- Product: Linkem (link emulator)



[Switch to Demo]

Summary/Conclusion

- libRIST compiles and runs on most platforms
- Code improvements and maintenance are ongoing, focused on efficiency, and meticulously tested
- As a FOSS project, libRIST benefits a wide range of people, organizations and uses



libRIST

Thank You

SipRadius

VSF
VIDEO SERVICES FORUM