

# LibRIST TUN

## *Who Needs Exciting When It Just Works?*

*Sergio Ammirata, Ph.D.  
SipRadius*



# Boring Is Beautiful

- We present a new use of a new feature: TUN support.
  - Security camera feed and PTZ (point-tilt-zoom) software commands via RIST.
  - Demonstrates RIST can now be applied to every day production tasks.





udp/TS (sends from camera),  
and PTZ service listens for  
camera manipulation  
commands.

RIST sender

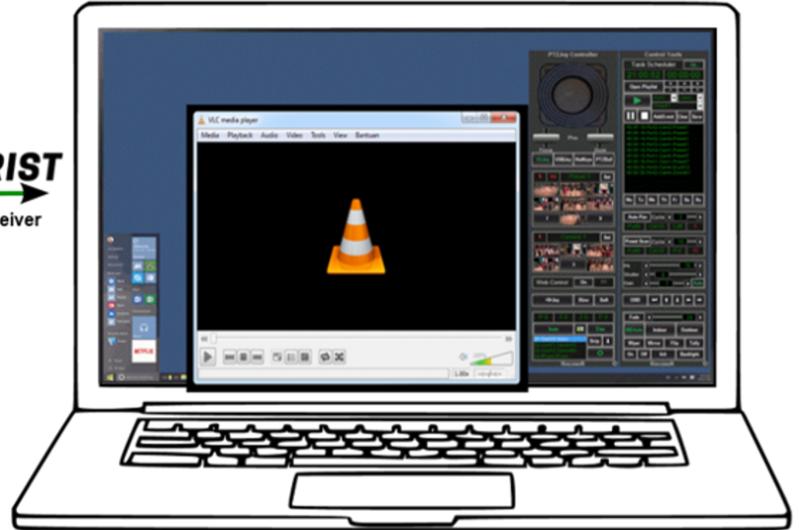


**lib** >>> **RIST**

**lib** >>> **RIST**

RIST receiver

libRIST exposes a TUN  
network bridge for PTZ  
control



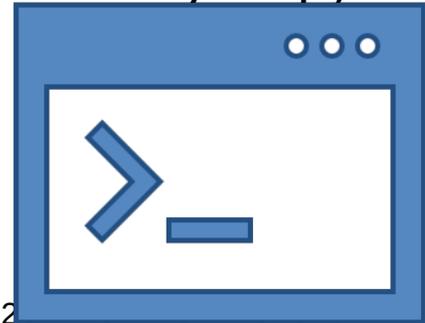
VLC plays video from rist receiver  
PTZ software controls camera.

# Just Two Command Line Options to Add

- -t tun# (your previously defined TUN device, as in -t tun11)
- -m # (0 for all protocols, 1 for all but udp, 2 for only udp)
- Example with TUN creation and proxy arp:

- ip tuntap add mode tun dev tun11
- ip addr add 192.168.101.102/24 dev tun11
- ip link set dev tun11 up
- (setup proxy arp)

```
ristsender -t tun11 -m 0 -i 'udp://127.0.0.1:12345' -o 'rist://@127.0.0.1:12345?cname=SENDER&bandwidth=10000&buffer-min=1000&buffer-max=1000&rtt-min=25&rtt-max=105&reorder-buffer=25&virt-dst-port=1968&weight=0&congestion-control=1&aes-type=128&secret=blarg&username=user01&password=RuXprE3d' -p 1 -v 6
```



# Command Line Help

```
johni@sfflinux:~
BALDOMERUS:/data/rist_test# ristsender --help
1632236187.68725710.01[INFO] Starting ristsender version: 0.2.6 libRIST library: v0.2.6-3-g1fa3eef-dirty API ver
sion: 4.1.1
1632236187.68733110.01[INFO] ristsender
Usage: %s [OPTIONS]
Where OPTIONS are:
  -i | --inputurl udp://... or rtp://... * | Comma separated list of input udp or rtp URLs
                                           | Use tun://@ to read udp data from a tun device defined
                                           | using the -t option
  -o | --outputurl rist://... * | Comma separated list of output rist URLs
  -b | --buffer value           | Default buffer size for packet retransmissions
  -s | --secret PWD             | Default pre-shared encryption secret
  -e | --encryption-type TYPE  | Default Encryption type (0, 128 = AES-128, 256 = AES-256)
  -p | --profile number         | Rist profile (0 = simple, 1 = main, 2 = advanced)
  -n | --null-packet-deletion   | Enable NPD, receiver needs to support this!
  -S | --statsinterval value (ms) | Interval at which stats get printed, 0 to disable
  -v | --verbose-level value    | To disable logging: -1, log levels match syslog levels
  -r | --remote-logging IP:PORT | Send logs and stats to this IP:PORT using udp messages
  -F | --srpfile filepath      | When in listening mode, use this file to hold the list
                               | of usernames and passwords to validate against. Use the
                               | ristsrppasswd tool to create the line entries.
  -t | --tun name               | Create a tun device and use it for data communications
  -m | --tun-mode number        | Data management on the tun interface:
                               | 0 = all data is accepted into and out of oob channel
                               | 1 = only non udp data is accepted (default)
                               | 2 = no data goes into or out of oob channel
  -f | --fast-start value       | Controls data output flow before handshake is completed
                               | 0 = hold data out
                               | 1 = start to send data immediately
  -h | --help                   | Show this help
  -u | --help-url               | Show all the possible url options
* == mandatory value
Default values: %s
  --profile 1
  --statsinterval 1000
  --verbose-level 6
version 0.2.6 libRIST library: v0.2.6-3-g1fa3eef-dirty API version: 4.1.1
BALDOMERUS:/data/rist_test#
```



# Demo



# Demo



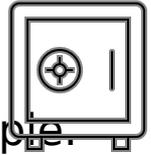
# What We Just Demonstrated

- Private IP Network Tunneling Through the Internet Has Always Been a Part of RIST
- The New TUN Feature Simply Exposes the Functionality to Other Applications Without Affecting the RIST Stream
- To One or More Applications, the Other Side of the TUN is Just Another Host Accessible by IP Address



# Why TUN is Important

- Userland Tunnel Across Internet Between Two Devices
- Secure, AES Encryption and libRIST Authorization Offers the Same Strong Security and Robust Error Correction to Communications Between Your Other Application(s).
  - Which is why we showed a security camera as a particular example.
- You Can Have Two-Way Communication Thru the TUN
  - Such as a chat app. There doesn't even have to be a relation between the media stream conveyed by RIST and applications using the TUN



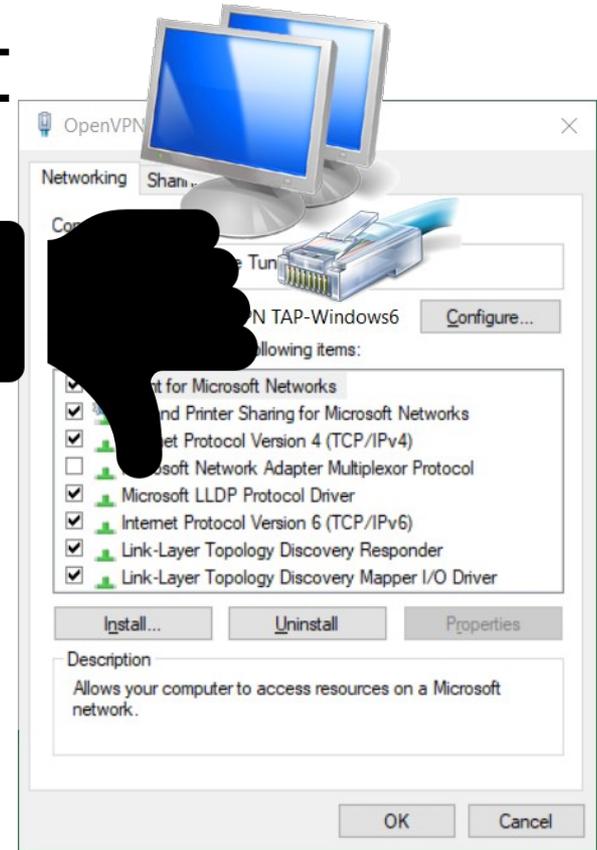
# Taking the Security Cam App Further

- Multiple Peer Support Could Add Geographic Flexibility
  - One TUN can support multiple cameras located at one remote
  - One PTZ app can support multiple sites via multiple peers
- EAP Authentication to peers combined with Robust AES Encryption With No Third Party Service Required



# One Caveat

- Linux or Mac OS Only Right Now.
- Though there *is* a TAP/TUN adapter widely available for Windows, it doesn't do IP networking in TUN mode!
- Solution: use a linux machine running rist as a proxy-arp bridge to your windows client.



# Summary

- It Just Works
- Highly Secure
- *The Very Fact That a RIST Video Demo Can Be This Boring Is a Tribute to RIST Maturity!*



Thank you!

